C 6

```
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMM         MMM     OOOOOOOOO     UUU         UUU  NNN         NNN  TTTTTTTTTTTTTTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMMMMM   MMMMMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM   MMM   MMM  OOO         OOO  UUU         UUU  NNNNNN      NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN   NNN   NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN      NNNNNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM  OOO         OOO  UUU         UUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
MMM         MMM     OOOOOOOOO     UUUUUUUUUUUUUUU  NNN         NNN        TTT
```

```
**FILE**ID**MOUNTIMG                              K  4










MM      MM    OOOOOO    UU      UU  NN      NN  TTTTTTTTTT    IIIIII    MM      MM    GGGGGGGG
MM      MM    OOOOOO    UU      UU  NN      NN  TTTTTTTTTT    IIIIII    MM      MM    GGGGGGGG
MMMM  MMMM  OO      OO  UU      UU  NN      NN      TT          II      MMMM  MMMM  GG
MMMM  MMMM  OO      OO  UU      UU  NN      NN      TT          II      MMMM  MMMM  GG
MM  MM  MM  OO      OO  UU      UU  NNNN    NN      TT          II      MM  MM  MM  GG
MM  MM  MM  OO      OO  UU      UU  NNNN    NN      TT          II      MM  MM  MM  GG
MM      MM  OO      OO  UU      UU  NN  NN  NN      TT          II      MM      MM  GG
MM      MM  OO      OO  UU      UU  NN  NN  NN      TT          II      MM      MM  GG
MM      MM  OO      OO  UU      UU  NN    NNNN      TT          II      MM      MM  GG  GGGGGG
MM      MM  OO      OO  UU      UU  NN    NNNN      TT          II      MM      MM  GG  GGGGGG
MM      MM  OO      OO  UU      UU  NN      NN      TT          II      MM      MM  GG      GG      ....
MM      MM  OO      OO  UU      UU  NN      NN      TT          II      MM      MM  GG      GG      ....
MM      MM    OOOOOO    UUUUUUUUUU  NN      NN      TT        IIIIII    MM      MM    GGGGGG        ....
MM      MM    OOOOOO    UUUUUUUUUU  NN      NN      TT        IIIIII    MM      MM    GGGGGG        ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

MOU
V04

54

```
0 MODULE MOUNTIMG (
0                 MAIN = PARSE_COMMAND,
0                 ADDRESSING_MODE (EXTERNAL = GENERAL),
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-006'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MOUNT Utility Structure Level 1
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains the data base and utilities used to acquire the
1 !       MOUNT command line from the CLI parser.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  29-Sep-1977  16:58
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-018 HH0044          Hai Huang               09-Aug-1984
1 !               Correctly parse /CACHE options.
1 !
1 !       V03-017 HH0041          Hai Huang               24-Jul-1984
```

```
1 !              Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
1 !
1 !      V03-016 DAS0003          David Solomon           09-Jul-1984
1 !              Add support for /NOREBUILD.
1 !
1 !      V03-015 HH0028           Hai Huang               27-Jun-1984
1 !              Make several qualifiers negatable (/CLUSTER, /GROUP,
1 !              /SYSTEM).
1 !
1 !      V03-014 HH0004           Hai Huang               09-Mar-1984
1 !              Add cluster-wide mount support.
1 !
1 !      V03-013 WMC0001          Wayne Cardoza           16-Jan-1984
1 !              Disable all journaling qualifiers.
1 !
1 !      V03-012 MCN0141          Maria del C. Nasr       27-Dec-1983
1 !              Add VALCNVERR message, and eliminate PARSE_ERROR routine
1 !              since it is not needed with new CLI interface.
1 !
1 !      V03-011 DAS0002          David Solomon           09-Dec-1983
1 !              Fix symbol name that was too long.
1 !
1 !      V03-010 DAS0001          David Solomon           29-Nov-1983
1 !              Add support for specifying maximum journal record size
1 !              with a new keyword, /JOURNAL=(RECORD_SIZE=n).
1 !
1 !      V03-009 MCN0138          Maria del C. Nasr       21-Nov-1983
1 !              Turn of NEWJOURNAL when /NOJOURNAL is specified.
1 !
1 !      V03-008 MCN0137          Maria del C. Nasr       12-Jul-1983
1 !              Change to new CLI interface.
1 !
1 !      V03-007 LMP0140          L. Mark Pilant          22-Aug-1983
1 !              Add support for alphanumeric UICs.
1 !
1 !      V03-006 MMD0188          Meg Dumont,     7-Jul-1983  10:00
1 !              Make the default for AVL/AVR the same from the DCL call
1 !              and from the system service call.
1 !
1 !      V03-005 MMD0116          Meg Dumont,     29-Mar-1983  0:40
1 !              Add support for AVL, AVR and new VMS prot on tape
1 !
1 !      V03-004 STJ49203         Steven T. Jeffreys,     08-Feb-1982
1 !              Set MNT$V_OVR_SETID if /OVERRIDE=SETID was specified.
1 !
1 !      V03-003 STJ0318          Steven T. Jeffreys,     15-Aug-1982
1 !              Added support for the journalling qualifiers.
1 !
1 !      V03-002 STJ0303          Steven T. Jeffreys,     18-May-1982
1 !              Replace the obsolete /UNLOCK qualifier with the /UNLOAD
1 !              qualifier.
1 !
1 !      V03-001 STJ0239          Steven T. Jeffreys,     17-Mar-1982
1 !              Relax the parsing restrictions on the device name
1 !              as specified in the /PROCESSOR=SAME:<device name>
1 !              qualifier.  Specifically, if no ":" is specified in
1 !              the device name, put one there.
```

```
;  115      0115  1 !
;  116      0116  1 !       V02-016 STJ0226         Steven T. Jeffreys,     17-Feb-1982
;  117      0117  1 !               Do not set the inhibit bit in the final status code.
;  118      0118  1 !               This effectively undoes edit #14.
;  119      0119  1 !
;  120      0120  1 !       V02-015 STJ0213         Steven T. Jeffreys,     11-Feb-1982
;  121      0121  1 !               Add support for the /COMMENT qualifier.
;  122      0122  1 !
;  123      0123  1 !       V02-014 STJ0201         Steven T. Jeffreys,     04-Feb-1982
;  124      0124  1 !               Set the inhibit bit in the final status code.
;  125      0125  1 !
;  126      0126  1 !       V02-013 STJ0187         Steven T. Jeffreys,     25-Jan-1982
;  127      0127  1 !               Changed MNT$V_MOUNTVER to MNT$V_NOMNTVER.
;  128      0128  1 !
;  129      0129  1 !       V02-012 STJ0172         Steven T. Jeffreys,     08-Jan-1982
;  130      0130  1 !               Changed $MOUNT interface to use *new* item list format.
;  131      0131  1 !
;  132      0132  1 !       V02-011 STJ0162         Steven T. Jeffreys,     04-Jan-1982
;  133      0133  1 !               Added support for the /OVERRIDE=LOCK, /NOCACHE, /MOUNTVER,
;  134      0134  1 !               and /MESSAGE qualifiers.
;  135      0135  1 !
;  136      0136  1 !       V02-010 STJ0153         Steven T. Jeffreys,     02-Jan-1981
;  137      0137  1 !               Extensive rewrite to support the $MOUNT system service.
;  138      0138  1 !
;  139      0139  1 !       V02-009 STJ0147         Steven T. jeffreys,     01-Dec-1981
;  140      0140  1 !               Fixed TPARSE table for /PROCESSOR= option.
;  141      0141  1 !
;  142      0142  1 !       V02-008 STJ0137         Steven T. Jeffreys,     02-Nov-1981
;  143      0143  1 !               Convert the command line parser to a separate image,
;  144      0144  1 !               which will parse the command line and then call the
;  145      0145  1 !               $MOUNT system service to complete the mount.
;  146      0146  1 !
;  147      0147  1 !       V02-007 STJ0036         Steven T. Jeffreys,     11-May-1981
;  148      0148  1 !               Added support for /ASSIST qualifier.
;  149      0149  1 !
;  150      0150  1 !       V02-006 ACG0167         Andrew C. Goldstein,    18-Apr-1980  13:38
;  151      0151  1 !               Previous revision history moved to MOUNT.REV
;  152      0152  1 !**
;  153      0153  1
;  154      0154  1
;  155      0155  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;  156      0156  1 REQUIRE 'SRC$:MOUDEF.B32';
;  157      0688  1 REQUIRE 'LIBD$:[VMSLIB.OBJ]INITMSG.REQ';
;  158      0820  1 LIBRARY 'SYS$LIBRARY:CLIMAC.L32';
;  159      0821  1 LIBRARY 'SYS$LIBRARY:TPAMAC.L32';
```

```
1
1  FORWARD ROUTINE
1          CACHE_ACT         : NOVALUE,
1          DATACHECK_ACT     : NOVALUE,
1          DENSITY_ACT       : NOVALUE,
1          GET_DEVICE        : NOVALUE,
1          GET_LABEL         : NOVALUE,
1          GET_LOG_NAME      : NOVALUE,
1          INITIALIZE_ACT    : NOVALUE,
1          JOURNAL_ACT       : NOVALUE,
1          OVERRIDE_ACT      : NOVALUE,
1          OWNER_UIC_ACT     : NOVALUE,
1          PARSE_QUALIFIER   : NOVALUE,
1          PROCESSOR_ACT     : NOVALUE,
1          PROTECTION_ACT    : NOVALUE,
1          MAIN_HANDLER,
1          BUILD_LIST        : NOVALUE;
1
1  !+
1  !
1  ! Impure data area. This area contains the MOUNT parameters extracted from
1  ! the command line by the associated parsing routines.
1  !
1  !-
1
1
1  OWN
1      DEVICE_COUNT,                                  ! number of devices specified
1      LABEL_COUNT,                                   ! number of volume labels specified
1      DEVICE_STRING          : VECTOR [DEVMAX*2],     ! descriptors of device name strings
1      LABEL_STRING           : VECTOR [LABMAX*2],     ! descriptors of volume label strings
1      LOG_NAME               : BBLOCK [DSC$C_S_BLN],  ! descriptor of logical name string
1      MOUNT_OPTIONS          : BITVECTOR [64],        ! option flags
1      MOUNT_FLAGS            : BBLOCK [4],            ! mount option flags for service
1
1      ! Value of qualifiers
1      !
1      ACCESS,                                        ! value of /ACCESSED qualifier
1      ACP_STRING             : BBLOCK [DSC$C_S_BLN],  ! descriptor of ACP device or name string
1      BLOCKSZ,                                       ! value of /BLOCKSIZE qualifier
1      EXT_CACHE,                                     ! space to allocate for extent cache
1      FID_CACHE,                                     ! space to allocate for file ID cache
1      QUO_CACHE,                                     ! space to allocate for quota cache
1      COMMENT_STRING         : BBLOCK [DSC$C_S_BLN],  ! descriptor of /COMMENT string
1      DENSITY,                                       ! value of /DENSITY qualifier
1      EXTENSION,                                     ! value of /EXTENSION qualifier
1      JRNL_QUOTA,                                    ! value of /JOURNAL=QUOTA keyword
1      JRNL_EXTEND,                                   ! value of /JOURNAL=EXTEND keyword
1      JRNL_SIZE,                                     ! value of /JOURNAL=SIZE keyword
1      JRNL_RECORD_SIZE,                              ! value of /JOURNAL=RECORD_SIZE keyword
1      OWNER_UIC,                                     ! value of /OWNER_UIC qualifier
1      PROTECTION,                                    ! value of /PROTECTION qualifier
1      RECORDSZ,                                      ! value of /RECORDSZ qualifier
1      STRUCT_NAME            : BBLOCK [DSC$C_S_BLN],  ! descriptor of volume set name
1                                                     !  (value of /BIND qualifier)
1      WINDOW,                                        ! value of /WINDOWS qualifier
1
```

```
1           CLI_DESC                : BBLOCK [DSC$C_S_BLN], ! CLI work descriptor
1           EXT_LIMIT               : INITIAL (-1),         ! limit of disk free space to cache
1           TPARSE_BLOCK            : BBLOCK [TPA$K_LENGTH0]
1                                     INITIAL (TPA$K_COUNT0, TPA$M_BLANKS OR TPA$M_ABBREV),
1           UIC,
1           ZERO;                                           ! variable whose value is 0
1
1       LITERAL
1           ITEM_SIZE = 12,
1           NUMBER_OF_ITEMS = 18,
1           ITEM_LIST_SIZE = ((ITEM_SIZE * DEVMAX) * 2) + (NUMBER_OF_ITEMS * ITEM_SIZE) + 4;
1
1       ! Descriptors for qualifiers names, used while parsing command line.
1       !
1       BIND
1           ACCESSED_DESC           = $DESCRIPTOR('ACCESSED'),
1           ASSIST_DESC             = $DESCRIPTOR('ASSIST'),
1           AUTOMATIC_DESC          = $DESCRIPTOR('AUTOMATIC'),
1           BIND_DESC               = $DESCRIPTOR('BIND'),
1           BLOCK_DESC              = $DESCRIPTOR('BLOCKSIZE'),
1           CACHE_DESC              = $DESCRIPTOR('CACHE'),
1           CLUSTER_DESC            = $DESCRIPTOR('CLUSTER'),
1           COMMENT_DESC            = $DESCRIPTOR('COMMENT'),
1           DATA_DESC               = $DESCRIPTOR('DATA_CHECK'),
1           DENSITY_DESC            = $DESCRIPTOR('DENSITY'),
1           EXTENSION_DESC          = $DESCRIPTOR('EXTENSION'),
1           FOREIGN_DESC            = $DESCRIPTOR('FOREIGN'),
1           GROUP_DESC              = $DESCRIPTOR('GROUP'),
1           HDR3_DESC               = $DESCRIPTOR('HDR3'),
1           INITIALIZE_DESC         = $DESCRIPTOR('INITIALIZE'),
1           JOURNAL_DESC            = $DESCRIPTOR('JOURNAL'),
1           LABEL_DESC              = $DESCRIPTOR('LABEL'),
1           MESSAGE_DESC            = $DESCRIPTOR('MESSAGE'),
1           MOUNT_VER_DESC          = $DESCRIPTOR('MOUNT_VERIFICATION'),
1           NOLABEL_DESC            = $DESCRIPTOR('NOLABEL'),
1           OVERRIDE_DESC           = $DESCRIPTOR('OVERRIDE'),
1           OWNER_DESC              = $DESCRIPTOR('OWNER_UIC'),
1           PROCESSOR_DESC          = $DESCRIPTOR('PROCESSOR'),
1           PROTECTION_DESC         = $DESCRIPTOR('PROTECTION'),
1           QUOTA_DESC              = $DESCRIPTOR('QUOTA'),
1           REBUILD_DESC            = $DESCRIPTOR('REBUILD'),
1           RECORD_DESC             = $DESCRIPTOR('RECORDSIZE'),
1           SHARE_DESC              = $DESCRIPTOR('SHARE'),
1           SYSTEM_DESC             = $DESCRIPTOR('SYSTEM'),
1           UNLOAD_DESC             = $DESCRIPTOR('UNLOAD'),
1           WINDOW_DESC             = $DESCRIPTOR('WINDOWS'),
1           WRITE_DESC              = $DESCRIPTOR('WRITE');
1
1       ! CLI parsing routines
1       !
1       EXTERNAL ROUTINE
1           LIB$CVT_DTB,
1           STR$COPY_DX,
1           CLI$GET_VALUE,                          ! retreives qualifiers value
1           CLI$PRESENT;                            ! determines if qualifier appears in
1                                                   !  command
1       EXTERNAL LITERAL
```

```
;   275          0936  1       CLI$_ABSENT,
;   276          0937  1       CLI$_DEFAULTED,
;   277          0938  1       CLI$_NEGATED,
;   278          0939  1       CLI$_PRESENT;
;   279          0940  1
```

```
1 GLOBAL ROUTINE PARSE_COMMAND =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine parses the MOUNT command line by calling the CLI
1 !       result parse routines, and leaves the results in the global data
1 !       area.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       MOUNT_PARSE
1 !
1 ! INPUT PARAMETERS:
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       parser impure area on preceding pages
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2     ITEM_LIST           : BBLOCK [ITEM_LIST_SIZE],      ! Storage for item list
2     END_OF_LIST,                                        ! Pointer to end of item list
2     STATUS;
2
2 ! Enable the main condition handler.  The handler will ensure that
2 ! the return status will have the MOUNT facility code.
2 !
2
2 ENABLE  MAIN_HANDLER;
2
2 ! Initialize list for system service.
2 !
2 END_OF_LIST = ITEM_LIST;
2
2 ! Initialize result parsing.
2 !
2 ZERO = 0;
2 MOUNT_OPTIONS = MOUNT_OPTIONS+4 = 0;
2 MOUNT_OPTIONS[OPT_MESSAGE] = 1;
2 MOUNT_OPTIONS[OPT_NOSHARE] = 1;
2 MOUNT_OPTIONS[OPT_NOLABEL] = 1;
```

```
2 MOUNT_OPTIONS[OPT_NOQUOTA] = 1;
2 MOUNT_OPTIONS[OPT_NOHDR3] = 1;
2 MOUNT_OPTIONS[OPT_NOUNLOAD] = 1;
2
2 ! Initialize CLI descriptor
2 !
2 CH$FILL ( 0, DSC$C_S_BLN, CLI_DESC );
2 CLI_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
2
2 PARSE_QUALIFIER ();
2
2 ! Get device names
2 !
2 GET_DEVICE ();
2
2 ! Get volume labels
2 !
2 GET_LABEL ();
2
2 ! Get logical name
2 !
2 GET_LOG_NAME ();
2
2 ! If no label given, construct null label string
2 !
2
2 IF NOT .MOUNT_OPTIONS [OPT_LABEL]
2 THEN
3     BEGIN
3     LABEL_STRING [0] = 0;
3     LABEL_STRING [1] = LABEL_STRING [1];
2     END;
2
2 ! Create a counted list of the addresses of all device names descriptors
2 !
3 INCR J FROM 0 TO (.DEVICE_COUNT - 1)
2 DO
2     BUILD_LIST ( MNT$_DEVNAM,
2                 .DEVICE_STRING [.J*2],
2                 .DEVICE_STRING [(.J*2)+1],
2                 END_OF_LIST );
2
2 ! Create a counted list of the addresses of all volume name descriptors
2 !
3 INCR J FROM 0 TO (.LABEL_COUNT - 1)
2 DO
2     BUILD_LIST ( MNT$_VOLNAM,
2                 .LABEL_STRING [.J*2],
2                 .LABEL_STRING [(.J*2)+1],
2                 END_OF_LIST );
2
2 ! Set up the parameter addresses for all specified parameters
2 !
2 ! Process the LOGNAM parameter
2 !
2
2 IF .MOUNT_OPTIONS [OPT_LOG_NAME]
```

```
2 THEN
2     BUILD_LIST ( MNT$_LOGNAM,
2                 .LOG_NAME [DSC$W_LENGTH],
2                 .LOG_NAME [DSC$A_POINTER],
2                 END_OF_LIST );
2
2 ! Process the /ACCESSED qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_ACCESSED]
2 THEN
2     BUILD_LIST ( MNT$_ACCESSED, 4, ACCESS, END_OF_LIST );
2
2 ! Process the /BIND qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_BIND]
2 THEN
2     BUILD_LIST ( MNT$_VOLSET, .STRUCT_NAME[DSC$W_LENGTH],
2                .STRUCT_NAME[DSC$A_POINTER], END_OF_LIST );
2
2 ! Process the /BLOCKSIZE qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_BLOCKSIZE]
2 THEN
2     BUILD_LIST ( MNT$_BLOCKSIZE, 4, BLOCKSZ, END_OF_LIST );
2
2 ! Process the /CACHE=([NO]EXTENT) qualifier
2 !
2 IF .EXT_CACHE GTR 0
2 THEN
2     BEGIN
3     BUILD_LIST (MNT$_EXTENT, 4, EXT_CACHE, END_OF_LIST);
2     END;
2 IF .MOUNT_OPTIONS [OPT_NOEXT_C]
2 THEN
2     BUILD_LIST (MNT$_EXTENT, 4, ZERO, END_OF_LIST);
2
2 ! Process the /CACHE=([NO]FILE_ID) qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_NOFID_C]
2 THEN
2     FID_CACHE= 1;
2 IF .FID_CACHE GTR 0
2 THEN
2     BUILD_LIST (MNT$_FILEID, 4, FID_CACHE, END_OF_LIST);
2
2 ! Process the /CACHE=(LIMIT) qualifier
2 !
2 IF .EXT_LIMIT GTR -1
2 THEN
2     BUILD_LIST (MNT$_LIMIT, 4, EXT_LIMIT, END_OF_LIST);
2
2 ! Process the /CACHE=([NO]QUOTA) qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_NOQUO_C]
2 THEN
2     BUILD_LIST (MNT$_QUOTA, 4, ZERO, END_OF_LIST);
2 IF .QUO_CACHE GTR 0
```

```
2 THEN
2     BUILD_LIST (MNT$_QUOTA, 4, QUO_CACHE, END_OF_LIST);
2
2 ! Process the /COMMENT qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_COMMENT]
2 THEN
2     BUILD_LIST ( MNT$_COMMENT, .COMMENT_STRING[DSC$W_LENGTH],
2                 .COMMENT_STRING[DSC$A_POINTER], END_OF_LIST );
2
2 ! Process the /DENSITY qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_DENSITY]
2 THEN
2     BUILD_LIST (MNT$_DENSITY, 4, DENSITY, END_OF_LIST);
2
2 ! Process the /EXTENSION qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_EXTENSION]
2 THEN
2     BUILD_LIST ( MNT$_EXTENSION, 4, EXTENSION, END_OF_LIST );
2
2 ! Process the /JOURNAL qualifier options
2 !
2 IF .JRNL_SIZE NEQ 0
2 THEN
2     BUILD_LIST (MNT$_JRNLSIZE, 4, JRNL_SIZE, END_OF_LIST);
2
2 IF .JRNL_RECORD_SIZE NEQ 0
2 THEN
2     BUILD_LIST (MNT$_JRNLRECORD_SIZE, 4, JRNL_RECORD_SIZE, END_OF_LIST);
2
2 IF .JRNL_EXTEND NEQ 0
2 THEN
2     BUILD_LIST (MNT$_JRNLEXTEND, 4, JRNL_EXTEND, END_OF_LIST);
2
2 IF .JRNL_QUOTA NEQ 0
2 THEN
2     BUILD_LIST (MNT$_JRNLQUOTA, 4, JRNL_QUOTA, END_OF_LIST);
2
2
2 ! Process the /OWNER_UIC qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_OWNER_UIC]
2 THEN
2     BUILD_LIST (MNT$_OWNER, 4, OWNER_UIC, END_OF_LIST);
2
2 ! Process the /PROCESSOR qualifier
2 !
2 IF .MOUNT_OPTIONS [OPT_UNIQUEACP]
2   OR .MOUNT_OPTIONS [OPT_SAMEACP]
2     OR .MOUNT_OPTIONS [OPT_FILEACP]
2 THEN
2     BUILD_LIST ( MNT$_PROCESSOR, .ACP_STRING [DSC$W_LENGTH],
2                 .ACP_STRING [DSC$A_POINTER], END_OF_LIST);
2
2 ! Process the /PROTECTION qualifer
```

```
!
IF .MOUNT_OPTIONS [OPT_PROTECTION]
THEN
    BUILD_LIST (MNT$_VPROT, 4, PROTECTION, END_OF_LIST);

! Process the /RECORDIZE qualifier
!
IF .MOUNT_OPTIONS [OPT_RECORDSZ]
THEN
    BUILD_LIST ( MNT$_RECORDSIZ, 4, RECORDSZ, END_OF_LIST );

! Process the /WINDOW qualifier
!
IF .MOUNT_OPTIONS [OPT_WINDOW]
THEN
    BUILD_LIST ( MNT$_WINDOW, 4, WINDOW, END_OF_LIST );


! Set the MOUNT flags according to their counterparts in MOUNT_OPTIONS
!

MOUNT_FLAGS [MNT$V_CLUSTER] = .MOUNT_OPTIONS [OPT_CLUSTER];
MOUNT_FLAGS [MNT$V_FOREIGN] = .MOUNT_OPTIONS [OPT_FOREIGN] OR .MOUNT_OPTIONS [OPT_NOLABEL];
MOUNT_FLAGS [MNT$V_GROUP] = .MOUNT_OPTIONS [OPT_GROUP];
MOUNT_FLAGS [MNT$V_INIT_ALL] = .MOUNT_OPTIONS [OPT_INIT_ALL];
MOUNT_FLAGS [MNT$V_INIT_CONT] = .MOUNT_OPTIONS [OPT_INIT_CONT];
MOUNT_FLAGS [MNT$V_MESSAGE] = .MOUNT_OPTIONS [OPT_MESSAGE];
MOUNT_FLAGS [MNT$V_NEWJRNL] = .MOUNT_OPTIONS [OPT_NEWJRNL];
MOUNT_FLAGS [MNT$V_NOASSIST] = NOT .MOUNT_OPTIONS [OPT_ASSIST];
MOUNT_FLAGS [MNT$V_NOAUTO] = .MOUNT_OPTIONS [OPT_NOAUTO];
MOUNT_FLAGS [MNT$V_NOCACHE] = .MOUNT_OPTIONS [OPT_NOCACHE];
MOUNT_FLAGS [MNT$V_NODISKQ] = .MOUNT_OPTIONS [OPT_NOQUOTA];
MOUNT_FLAGS [MNT$V_NOHDR3] = .MOUNT_OPTIONS [OPT_NOHDR3];
MOUNT_FLAGS [MNT$V_NOJRNL] = .MOUNT_OPTIONS [OPT_NOJRNL];
MOUNT_FLAGS [MNT$V_NOMNTVER] = NOT .MOUNT_OPTIONS [OPT_MOUNTVER];
MOUNT_FLAGS [MNT$V_NOUNLOAD] = .MOUNT_OPTIONS [OPT_NOUNLOAD];
MOUNT_FLAGS [MNT$V_NOWRITE] = NOT .MOUNT_OPTIONS [OPT_WRITE];
MOUNT_FLAGS [MNT$V_OVR_ACCESS] = .MOUNT_OPTIONS [OPT_OVR_ACC];
MOUNT_FLAGS [MNT$V_OVR_EXP] = .MOUNT_OPTIONS [OPT_OVR_EXP];
MOUNT_FLAGS [MNT$V_OVR_IDENT] = .MOUNT_OPTIONS [OPT_OVR_ID];
MOUNT_FLAGS [MNT$V_OVR_LOCK] = .MOUNT_OPTIONS [OPT_OVR_LOCK];
MOUNT_FLAGS [MNT$V_OVR_SETID] = .MOUNT_OPTIONS [OPT_OVR_SETID];
MOUNT_FLAGS [MNT$V_OVR_VOLO] = .MOUNT_OPTIONS [OPT_OVR_VOLO];
MOUNT_FLAGS [MNT$V_READCHECK] = .MOUNT_OPTIONS [OPT_READCHECK];
MOUNT_FLAGS [MNT$V_SHARE] = .MOUNT_OPTIONS [OPT_SHARE];
MOUNT_FLAGS [MNT$V_SYSTEM] = .MOUNT_OPTIONS [OPT_SYSTEM];
MOUNT_FLAGS [MNT$V_WRITECHECK] = .MOUNT_OPTIONS [OPT_WRITECHECK];
MOUNT_FLAGS [MNT$V_WRITETHRU] = .MOUNT_OPTIONS [OPT_WTHRU];
MOUNT_FLAGS [MNT$V_NOREBUILD] = .MOUNT_OPTIONS [OPT_NOREBUILD];

! Build an item list entry for mount flags, then terminate the item list
! with a zero value.
!
BUILD_LIST ( MNT$_FLAGS, 4, MOUNT_FLAGS, END_OF_LIST );
.END_OF_LIST = 0;

! Now that all the parameters have been parsed, call the $MOUNT system service.
```

```
2 ! Note the informational messages may be issued from mount via a $PUTMSG and
2 ! a status value from the call will be returned as well.
2 !
2 STATUS = $MOUNT (ITMLST = ITEM_LIST);          ! Mount the volume(s)
2
2 RETURN (.STATUS)                               ! Return status of $MOUNT call
2
1 END;                                           ! end of routine PARSE_COMMAND

                                                   .TITLE  MOUNTIMG
                                                   .IDENT  \V04-000\

                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

          44  45  53  53  45  43  43  41  00000 P.AAB:   .ASCII  \ACCESSED\
                               00000008.  00008 P.AAA:   .LONG   8
                               00000000'  0000C          .ADDRESS P.AAB
                      54  53  49  53  53  41  00010 P.AAD:   .ASCII  \ASSIST\
                                          00016          .BLKB   2
                               00000006.  00018 P.AAC:   .LONG   6
                               00000000'  0001C          .ADDRESS P.AAD
      43  49  54  41  4D  4F  54  55  41  00020 P.AAF:   .ASCII  \AUTOMATIC\
                                          00029          .BLKB   3
                               00000009.  0002C P.AAE:   .LONG   9
                               00000000'  00030          .ADDRESS P.AAF
                              44  4E  49  42  00034 P.AAH:   .ASCII  \BIND\
                               00000004.  00038 P.AAG:   .LONG   4
                               00000000'  0003C          .ADDRESS P.AAH
      45  5A  49  53  4B  43  4F  4C  42  00040 P.AAJ:   .ASCII  \BLOCKSIZE\
                                          00049          .BLKB   3
                               00000009.  0004C P.AAI:   .LONG   9
                               00000000'  00050          .ADDRESS P.AAJ
                          45  48  43  41  43  00054 P.AAL:   .ASCII  \CACHE\
                                          00059          .BLKB   3
                               00000005.  0005C P.AAK:   .LONG   5
                               00000000'  00060          .ADDRESS P.AAL
              52  45  54  53  55  4C  43  00064 P.AAN:   .ASCII  \CLUSTER\
                                          0006B          .BLKB   1
                               00000007.  0006C P.AAM:   .LONG   7
                               00000000'  00070          .ADDRESS P.AAN
              54  4E  45  4D  4D  4F  43  00074 P.AAP:   .ASCII  \COMMENT\
                                          0007B          .BLKB   1
                               00000007.  0007C P.AAO:   .LONG   7
                               00000000'  00080          .ADDRESS P.AAP
  4B  43  45  48  43  5F  41  54  41  44  00084 P.AAR:   .ASCII  \DATA_CHECK\
                                          0008E          .BLKB   2
                               0000000A.  00090 P.AAQ:   .LONG   10
                               00000000'  00094          .ADDRESS P.AAR
              59  54  49  53  4E  45  44  00098 P.AAT:   .ASCII  \DENSITY\
                                          0009F          .BLKB   1
                               00000007.  000A0 P.AAS:   .LONG   7
                               00000000'  000A4          .ADDRESS P.AAT
      4E  4F  49  53  4E  45  54  58  45  000A8 P.AAV:   .ASCII  \EXTENSION\
                                          000B1          .BLKB   3
                               00000009.  000B4 P.AAU:   .LONG   9
                               00000000'  000B8          .ADDRESS P.AAV
```

```
                         4E 47 49 45 52 4F 46  000BC P.AAX:  .ASCII   \FOREIGN\                       ;
                                               000C3         .BLKB    1
                                   00000007.   000C4 P.AAW:  .LONG    7                               ;
                                   00000000'   000C8         .ADDRESS P.AAX                           ;
                               50 55 4F 52 47  000CC P.AAZ:  .ASCII   \GROUP\                         ;
                                               000D1         .BLKB    3
                                   00000005.   000D4 P.AAY:  .LONG    5                               ;
                                   00000000'   000D8         .ADDRESS P.AAZ                           ;
                                  33 52 44 48  000DC P.ABB:  .ASCII   \HDR3\                          ;
                                   00000004.   000E0 P.ABA:  .LONG    4                               ;
                                   00000000'   000E4         .ADDRESS P.ABB                           ;
                45 5A 49 4C 41 49 54 49 4E 49  000E8 P.ABD:  .ASCII   \INITIALIZE\                    ;
                                               000F2         .BLKB    2
                                   0000000A.   000F4 P.ABC:  .LONG    10                              ;
                                   00000000'   000F8         .ADDRESS P.ABD                           ;
                         4C 41 4E 52 55 4F 4A  000FC P.ABF:  .ASCII   \JOURNAL\                       ;
                                               00103         .BLKB    1
                                   00000007.   00104 P.ABE:  .LONG    7                               ;
                                   00000000'   00108         .ADDRESS P.ABF                           ;
                               4C 45 42 41 4C  0010C P.ABH:  .ASCII   \LABEL\                         ;
                                               00111         .BLKB    3
                                   00000005.   00114 P.ABG:  .LONG    5                               ;
                                   00000000'   00118         .ADDRESS P.ABH                           ;
                         45 47 41 53 53 45 4D  0011C P.ABJ:  .ASCII   \MESSAGE\                       ;
                                               00123         .BLKB    1
                                   00000007.   00124 P.ABI:  .LONG    7                               ;
                                   00000000'   00128         .ADDRESS P.ABJ                           ;
54 41 43 49 46 49 52 45 56 5F 54 4E 55 4F 4D  0012C P.ABL:  .ASCII   \MOUNT_VERIFICATION\            ;
                                     4E 4F 49  0013B                                                  ;
                                               0013E         .BLKB    2
                                   00000012.   00140 P.ABK:  .LONG    18                              ;
                                   00000000'   00144         .ADDRESS P.ABL                           ;
                         4C 45 42 41 4C 4F 4E  00148 P.ABN:  .ASCII   \NOLABEL\                       ;
                                               0014F         .BLKB    1
                                   00000007.   00150 P.ABM:  .LONG    7                               ;
                                   00000000'   00154         .ADDRESS P.ABN                           ;
                      45 44 49 52 52 45 56 4F  00158 P.ABP:  .ASCII   \OVERRIDE\                      ;
                                   00000008.   00160 P.ABO:  .LONG    8                               ;
                                   00000000'   00164         .ADDRESS P.ABP                           ;
                   43 49 55 5F 52 45 4E 57 4F  00168 P.ABR:  .ASCII   \OWNER_UIC\                     ;
                                               00171         .BLKB    3
                                   00000009.   00174 P.ABQ:  .LONG    9                               ;
                                   00000000'   00178         .ADDRESS P.ABR                           ;
                   52 4F 53 53 45 43 4F 52 50  0017C P.ABT:  .ASCII   \PROCESSOR\                     ;
                                               00185         .BLKB    3
                                   00000009.   00188 P.ABS:  .LONG    9                               ;
                                   00000000'   0018C         .ADDRESS P.ABT                           ;
                4E 4F 49 54 43 45 54 4F 52 50  00190 P.ABV:  .ASCII   \PROTECTION\                    ;
                                               0019A         .BLKB    2
                                   0000000A.   0019C P.ABU:  .LONG    10                              ;
                                   00000000'   001A0         .ADDRESS P.ABV                           ;
                               41 54 4F 55 51  001A4 P.ABX:  .ASCII   \QUOTA\                         ;
                                               001A9         .BLKB    3
                                   00000005.   001AC P.ABW:  .LONG    5                               ;
                                   00000000'   001B0         .ADDRESS P.ABX                           ;
                         44 4C 49 55 42 45 52  001B4 P.ABZ:  .ASCII   \REBUILD\                       ;
                                               001BB         .BLKB    1
```

```
                              00000007  001BC P.ABY:  .LONG    7                      ;
                              00000000' 001C0         .ADDRESS P.ABZ                  ;
45  5A  49  53  44  52  4F  43  45  52  001C4 P.ACB:  .ASCII   \RECORDSIZE\           ;
                                        001CE         .BLKB    2
                              0000000A  001D0 P.ACA:  .LONG    10                     ;
                              00000000' 001D4         .ADDRESS P.ACB                  ;
                    45  52  41  48  53  001D8 P.ACD:  .ASCII   \SHARE\                ;
                                        001DD         .BLKB    3
                              00000005  001E0 P.ACC:  .LONG    5                      ;
                              00000000' 001E4         .ADDRESS P.ACD                  ;
                4D  45  54  53  59  53  001E8 P.ACF:  .ASCII   \SYSTEM\               ;
                                        001EE         .BLKB    2
                              00000006  001F0 P.ACE:  .LONG    6                      ;
                              00000000' 001F4         .ADDRESS P.ACF                  ;
                44  41  4F  4C  4E  55  001F8 P.ACH:  .ASCII   \UNLOAD\               ;
                                        001FE         .BLKB    2
                              00000006  00200 P.ACG:  .LONG    6                      ;
                              00000000' 00204         .ADDRESS P.ACH                  ;
            53  57  4F  44  4E  49  57  00208 P.ACJ:  .ASCII   \WINDOWS\              ;
                                        0020F         .BLKB    1
                              00000007  00210 P.ACI:  .LONG    7                      ;
                              00000000' 00214         .ADDRESS P.ACJ                  ;
                    45  54  49  52  57  00218 P.ACL:  .ASCII   \WRITE\                ;
                                        0021D         .BLKB    3
                              00000005  00220 P.ACK:  .LONG    5                      ;
                              00000000' 00224         .ADDRESS P.ACL                  ;

                                                      .PSECT   $OWN$,NOEXE,2

                                        00000 DEVICE_COUNT:
                                                      .BLKB    4
                                        00004 LABEL_COUNT:
                                                      .BLKB    4
                                        00008 DEVICE_STRING:
                                                      .BLKB    128
                                        00088 LABEL_STRING:
                                                      .BLKB    128
                                        00108 LOG_NAME:
                                                      .BLKB    8
                                        00110 MOUNT_OPTIONS:
                                                      .BLKB    8
                                        00118 MOUNT_FLAGS:
                                                      .BLKB    4
                                        0011C ACCESS: .BLKB    4
                                        00120 ACP_STRING:
                                                      .BLKB    8
                                        00128 BLOCKSZ:.BLKB    4
                                        0012C EXT_CACHE:
                                                      .BLKB    4
                                        00130 FID_CACHE:
                                                      .BLKB    4
                                        00134 QUO_CACHE:
                                                      .BLKB    4
                                        00138 COMMENT_STRING:
                                                      .BLKB    8
                                        00140 DENSITY:.BLKB    4
                                        00144 EXTENSION:
```

```
                                                  .BLKB   4
                              00148 JRNL_QUOTA:
                                                  .BLKB   4
                              0014C JRNL_EXTEND:
                                                  .BLKB   4
                              00150 JRNL_SIZE:
                                                  .BLKB   4
                              00154 JRNL_RECORD_SIZE:
                                                  .BLKB   4
                              00158 OWNER_UIC:
                                                  .BLKB   4
                              0015C PROTECTION:
                                                  .BLKB   4
                              00160 RECORDSZ:
                                                  .BLKB   4
                              00164 STRUCT_NAME:
                                                  .BLKB   8
                              0016C WINDOW:       .BLKB   4
                              00170 CLI_DESC:
                                                  .BLKB   8
                    FFFFFFFF  00178 EXT_LIMIT:
                                                  .LONG   -1                                               ;
          00000003  00000008  0017C TPARSE_BLOCK:
                                                  .LONG   8, 3                                             ;
                              00184               .BLKB   28
                              001A0 UIC:          .BLKB   4
                              001A4 ZERO:         .BLKB   4

                                    ACCESSED_DESC=         P.AAA
                                    ASSIST_DESC=           P.AAC
                                    AUTOMATIC_DESC=        P.AAE
                                    BIND_DESC=             P.AAG
                                    BLOCK_DESC=            P.AAI
                                    CACHE_DESC=            P.AAK
                                    CLUSTER_DESC=          P.AAM
                                    COMMENT_DESC=          P.AAO
                                    DATA_DESC=             P.AAQ
                                    DENSITY_DESC=          P.AAS
                                    EXTENSION_DESC=        P.AAU
                                    FOREIGN_DESC=          P.AAW
                                    GROUP_DESC=            P.AAY
                                    HDR3_DESC=             P.ABA
                                    INITIALIZE_DESC=       P.ABC
                                    JOURNAL_DESC=          P.ABE
                                    LABEL_DESC=            P.ABG
                                    MESSAGE_DESC=          P.ABI
                                    MOUNT_VER_DESC=        P.ABK
                                    NOLABEL_DESC=          P.ABM
                                    OVERRIDE_DESC=         P.ABO
                                    OWNER_DESC=            P.ABQ
                                    PROCESSOR_DESC=        P.ABS
                                    PROTECTION_DESC=       P.ABU
                                    QUOTA_DESC=            P.ABW
                                    REBUILD_DESC=          P.ABY
                                    RECORD_DESC=           P.ACA
                                    SHARE_DESC=            P.ACC
                                    SYSTEM_DESC=           P.ACE
```

```
                                                             UNLOAD_DESC=     P.ACG
                                                             WINDOW_DESC=     P.ACI
                                                             WRITE_DESC=      P.ACK
                                                                  .EXTRN   LIB$CVT_DTB, STR$COPY_DX
                                                                  .EXTRN   CLI$GET_VALUE, CLI$PRESENT
                                                                  .EXTRN   CLI$_ABSENT, CLI$_DEFAULTED
                                                                  .EXTRN   CLI$_NEGATED, CLI$_PRESENT
                                                                  .EXTRN   SYS$MOUNT

                                                                  .PSECT   $CODE$,NOWRT,2

                                                  00FC 00000      .ENTRY   PARSE_COMMAND, Save R2,R3,R4,R5,R6,R7     ; 0941
                             57          0000V    CF  9E 00002      MOVAB    BUILD_LIST, R7                            ;
                             56          0000'    CF  9E 00007      MOVAB    MOUNT_FLAGS, R6                           ;
                             5E           FDA4    CE  9E 0000C      MOVAB    -604(SP), SP                              ;
                             6D           0391    CF  DE 00011      MOVAL    30$, (FP)                                 ; 0974
                                                  5E  DD 00016      PUSHL    SP                                        ; 0989
                                          008C    C6  D4 00018      CLRL     ZERO                                      ; 0993
                                            F8    A6  7C 0001C      CLRQ     MOUNT_OPTIONS                             ; 0994
                      FE     A6                   08  88 0001F      BISB2    #8, MOUNT_OPTIONS+6                       ; 0995
                      F8     A6           1010    8F  A8 00023      BISW2    #4112, MOUNT_OPTIONS                      ; 0997
                      FD     A6                   14  88 00029      BISB2    #20, MOUNT_OPTIONS+5                      ; 0999
                      F9     A6                   04  88 0002D      BISB2    #4, MOUNT_OPTIONS+1                       ; 1000
        08           00              6E                   00  2C 00031      MOVC5    #0, (SP), #0, #8, CLI_DESC                ; 1004
                                            58    A6      00036                                                ;
                      5B     A6                   02  90 00038      MOVB     #2, CLI_DESC+3                            ; 1005
                   0000V     CF                   00  FB 0003C      CALLS    #0, PARSE_QUALIFIER                       ; 1007
                   0000V     CF                   00  FB 00041      CALLS    #0, GET_DEVICE                            ; 1011
                   0000V     CF                   00  FB 00046      CALLS    #0, GET_LABEL                             ; 1015
                   0000V     CF                   00  FB 0004B      CALLS    #0, GET_LOG_NAME                          ; 1019
                                            FB    A6  95 00050      TSTB     MOUNT_OPTIONS+3                           ; 1024
                                                  0B  19 00053      BLSS     1$                                        ;
                                          FF70    C6  D4 00055      CLRL     LABEL_STRING                              ; 1027
                    FF74     C6           FF74    C6  9E 00059      MOVAB    LABEL_STRING+4, LABEL_STRING+4            ; 1028
                             53           FEE8    C6  D0 00060 1$:  MOVL     DEVICE_COUNT, R3                          ; 1033
                             52                   01  CE 00065      MNEGL    #1, J                                     ;
                                                  19  11 00068      BRB      3$                                        ;
                                                  5E  DD 0006A 2$:  PUSHL    SP                                        ; 1035
                  50         52                   01  78 0006C      ASHL     #1, J, R0                                 ; 1037
                                          FEF4  C640  DD 00070      PUSHL    DEVICE_STRING+4[R0]                       ;
                  50         52                   01  78 00075      ASHL     #1, J, R0                                 ; 1036
                                          FEF0  C640  DD 00079      PUSHL    DEVICE_STRING[R0]                         ;
                                                  01  DD 0007E      PUSHL    #1                                        ; 1035
                             67                   04  FB 00080      CALLS    #4, BUILD_LIST                            ;
                  E3         52                   53  F2 00083 3$:  AOBLSS   R3, J, 2$                                 ;
                             53           FEEC    C6  D0 00087      MOVL     LABEL_COUNT, R3                           ; 1042
                             52                   01  CE 0008C      MNEGL    #1, J                                     ;
                                                  19  11 0008F      BRB      5$                                        ;
                                                  5E  DD 00091 4$:  PUSHL    SP                                        ; 1044
                  50         52                   01  78 00093      ASHL     #1, J, R0                                 ; 1046
                                          FF74  C640  DD 00097      PUSHL    LABEL_STRING+4[R0]                        ;
                  50         52                   01  78 0009C      ASHL     #1, J, R0                                 ; 1045
                                          FF70  C640  DD 000A0      PUSHL    LABEL_STRING[R0]                          ;
                                                  02  DD 000A5      PUSHL    #2                                        ; 1044
                             67                   04  FB 000A7      CALLS    #4, BUILD_LIST                            ;
                  E3         52                   53  F2 000AA 5$:  AOBLSS   R3, J, 4$                                 ;
                  0E    FB   A6                   05  E1 000AE      BBC      #5, MOUNT_OPTIONS+3, 6$                   ; 1054
```

```
                              5E  DD 000B3          PUSHL   SP                                      : 1056
                          F4  A6  DD 000B5          PUSHL   LOG_NAME+4                              : 1058
                7E        F0  A6  3C 000B8          MOVZWL  LOG_NAME, -(SP)                         : 1057
                              03  DD 000BC          PUSHL   #3                                      : 1056
                67            04  FB 000BE          CALLS   #4, BUILD_LIST                          :
0C      FB      A6            01  E1 000C1 6$:      BBC     #1, MOUNT_OPTIONS+3, 7$                 : 1063
                              5E  DD 000C6          PUSHL   SP                                      : 1065
                          04  A6  9F 000C8          PUSHAB  ACCESS                                  :
                              04  DD 000CB          PUSHL   #4                                      :
                              05  DD 000CD          PUSHL   #5                                      :
                67            04  FB 000CF          CALLS   #4, BUILD_LIST                          :
                0E        FD  A6  E9 000D2 7$:      BLBC    MOUNT_OPTIONS+5, 8$                     : 1069
                              5E  DD 000D6          PUSHL   SP                                      : 1071
                          50  A6  DD 000D8          PUSHL   STRUCT_NAME+4                           : 1072
                7E        4C  A6  3C 000DB          MOVZWL  STRUCT_NAME, -(SP)                      : 1071
                              07  DD 000DF          PUSHL   #7                                      :
                67            04  FB 000E1          CALLS   #4, BUILD_LIST                          :
                0C        FA  A6  E9 000E4 8$:      BLBC    MOUNT_OPTIONS+2, 9$                     : 1076
                              5E  DD 000E8          PUSHL   SP                                      : 1078
                          10  A6  9F 000EA          PUSHAB  BLOCKSZ                                 :
                              04  DD 000ED          PUSHL   #4                                      :
                              08  DD 000EF          PUSHL   #8                                      :
                67            04  FB 000F1          CALLS   #4, BUILD_LIST                          :
                          14  A6  D5 000F4 9$:      TSTL    EXT_CACHE                               : 1082
                              0C  15 000F7          BLEQ    10$                                     :
                              5E  DD 000F9          PUSHL   SP                                      : 1085
                          14  A6  9F 000FB          PUSHAB  EXT_CACHE                               :
                              04  DD 000FE          PUSHL   #4                                      :
                              0A  DD 00100          PUSHL   #10                                     :
                67            04  FB 00102          CALLS   #4, BUILD_LIST                          :
                          FD  A6  95 00105 10$:     TSTB    MOUNT_OPTIONS+5                         : 1087
                              0D  18 00108          BGEQ    11$                                     :
                              5E  DD 0010A          PUSHL   SP                                      : 1089
                        008C  C6  9F 0010C          PUSHAB  ZERO                                    :
                              04  DD 00110          PUSHL   #4                                      :
                              0A  DD 00112          PUSHL   #10                                     :
                67            04  FB 00114          CALLS   #4, BUILD_LIST                          :
                04        FE  A6  E9 00117 11$:     BLBC    MOUNT_OPTIONS+6, 12$                    : 1093
        18      A6            01  D0 0011B          MOVL    #1, FID_CACHE                           : 1095
                          18  A6  D5 0011F 12$:     TSTL    FID_CACHE                               : 1096
                              0C  15 00122          BLEQ    13$                                     :
                              5E  DD 00124          PUSHL   SP                                      : 1098
                          18  A6  9F 00126          PUSHAB  FID_CACHE                               :
                              04  DD 00129          PUSHL   #4                                      :
                              0B  DD 0012B          PUSHL   #11                                     :
                67            04  FB 0012D          CALLS   #4, BUILD_LIST                          :
                          60  A6  D5 00130 13$:     TSTL    EXT_LIMIT                               : 1102
                              0C  19 00133          BLSS    14$                                     :
                              5E  DD 00135          PUSHL   SP                                      : 1104
                          60  A6  9F 00137          PUSHAB  EXT_LIMIT                               :
                              04  DD 0013A          PUSHL   #4                                      :
                              0C  DD 0013C          PUSHL   #12                                     :
                67            04  FB 0013E          CALLS   #4, BUILD_LIST                          :
0D      FE      A6            01  E1 00141 14$:     BBC     #1, MOUNT_OPTIONS+6, 15$                : 1108
                              5E  DD 00146          PUSHL   SP                                      : 1110
                        008C  C6  9F 00148          PUSHAB  ZERO                                    :
                              04  DD 0014C          PUSHL   #4                                      :
```

```
                            0F  DD 0014E         PUSHL   #15                              ;
                 67         04  FB 00150         CALLS   #4, BUILD_LIST                   ;
                       1C   A6  D5 00153  15$:   TSTL    QUO_CACHE                        ; 1111
                            0C  15 00156         BLEQ    16$                              ;
                            5E  DD 00158         PUSHL   SP                               ; 1113
                       1C   A6  9F 0015A         PUSHAB  QUO_CACHE                        ;
                            04  DD 0015D         PUSHL   #4                               ;
                            0F  DD 0015F         PUSHL   #15                              ;
                 67         04  FB 00161         CALLS   #4, BUILD_LIST                   ;
     0E    F8    A6         03  E1 00164  16$:   BBC     #3, MOUNT_OPTIONS, 17$           ; 1117
                            5E  DD 00169         PUSHL   SP                               ; 1119
                       24   A6  DD 0016B         PUSHL   COMMENT_STRING+4                 ; 1120
                 7E    20   A6  3C 0016E         MOVZWL  COMMENT_STRING, -(SP)            ; 1119
                            14  DD 00172         PUSHL   #20                              ;
                 67         04  FB 00174         CALLS   #4, BUILD_LIST                   ;
                 0C    F8   A6  E9 00177  17$:   BLBC    MOUNT_OPTIONS, 18$               ; 1124
                            5E  DD 0017B         PUSHL   SP                               ; 1126
                       28   A6  9F 0017D         PUSHAB  DENSITY                          ;
                            04  DD 00180         PUSHL   #4                               ;
                            09  DD 00182         PUSHL   #9                               ;
                 67         04  FB 00184         CALLS   #4, BUILD_LIST                   ;
                       FA   A6  95 00187  18$:   TSTB    MOUNT_OPTIONS+2                  ; 1130
                            0C  18 0018A         BGEQ    19$                              ;
                            5E  DD 0018C         PUSHL   SP                               ; 1132
                       2C   A6  9F 0018E         PUSHAB  EXTENSION                        ;
                            04  DD 00191         PUSHL   #4                               ;
                            12  DD 00193         PUSHL   #18                              ;
                 67         04  FB 00195         CALLS   #4, BUILD_LIST                   ;
                       38   A6  D5 00198  19$:   TSTL    JRNL_SIZE                        ; 1136
                            0C  13 0019B         BEQL    20$                              ;
                            5E  DD 0019D         PUSHL   SP                               ; 1138
                       38   A6  9F 0019F         PUSHAB  JRNL_SIZE                        ;
                            04  DD 001A2         PUSHL   #4                               ;
                            15  DD 001A4         PUSHL   #21                              ;
                 67         04  FB 001A6         CALLS   #4, BUILD_LIST                   ;
                       3C   A6  D5 001A9  20$:   TSTL    JRNL_RECORD_SIZE                 ; 1140
                            0C  13 001AC         BEQL    21$                              ;
                            5E  DD 001AE         PUSHL   SP                               ; 1142
                       3C   A6  9F 001B0         PUSHAB  JRNL_RECORD_SIZE                 ;
                            04  DD 001B3         PUSHL   #4                               ;
                            18  DD 001B5         PUSHL   #24                              ;
                 67         04  FB 001B7         CALLS   #4, BUILD_LIST                   ;
                       34   A6  D5 001BA  21$:   TSTL    JRNL_EXTEND                      ; 1144
                            0C  13 001BD         BEQL    22$                              ;
                            5E  DD 001BF         PUSHL   SP                               ; 1146
                       34   A6  9F 001C1         PUSHAB  JRNL_EXTEND                      ;
                            04  DD 001C4         PUSHL   #4                               ;
                            16  DD 001C6         PUSHL   #22                              ;
                 67         04  FB 001C8         CALLS   #4, BUILD_LIST                   ;
                       30   A6  D5 001CB  22$:   TSTL    JRNL_QUOTA                       ; 1148
                            0C  13 001CE         BEQL    23$                              ;
                            5E  DD 001D0         PUSHL   SP                               ; 1150
                       30   A6  9F 001D2         PUSHAB  JRNL_QUOTA                       ;
                            04  DD 001D5         PUSHL   #4                               ;
                            17  DD 001D7         PUSHL   #23                              ;
                 67         04  FB 001D9         CALLS   #4, BUILD_LIST                   ;
     0C    FA    A6         02  E1 001DC  23$:   BBC     #2, MOUNT_OPTICNS+2, 24$         ; 1155
```

```
                                 5E  DD 001E1         PUSHL    SP                                     ; 1157
                             40  A6  9F 001E3         PUSHAB   OWNER_UIC                              ;
                                 04  DD 001E6         PUSHL    #4                                     ;
                                 0D  DD 001E8         PUSHL    #13                                    ;
                        67       04  FB 001EA         CALLS    #4, BUILD_LIST                         ;
               0A   FB  A6       02  E0 001ED 24$:   BBS      #2, MOUNT_OPTIONS+3, 25$               ; 1161
               05   FB  A6       03  E0 001F2         BBS      #3, MOUNT_OPTIONS+3, 25$               ; 1162
               0E   FB  A6       04  E1 001F7         BBC      #4, MOUNT_OPTIONS+3, 26$               ; 1163
                                 5E  DD 001FC 25$:   PUSHL    SP                                     ; 1165
                             0C  A6  DD 001FE         PUSHL    ACP_STRING+4                           ; 1166
                        7E   08  A6  3C 00201         MOVZWL   ACP_STRING, -(SP)                      ; 1165
                                 06  DD 00205         PUSHL    #6                                     ;
                        67       04  FB 00207         CALLS    #4, BUILD_LIST                         ;
               0C   FA  A6       01  E1 0020A 26$:   BBC      #1, MOUNT_OPTIONS+2, 27$               ; 1170
                                 5E  DD 0020F         PUSHL    SP                                     ; 1172
                             44  A6  9F 00211         PUSHAB   PROTECTION                             ;
                                 04  DD 00214         PUSHL    #4                                     ;
                                 0E  DD 00216         PUSHL    #14                                    ;
                        67       04  FB 00218         CALLS    #4, BUILD_LIST                         ;
               0C   FC  A6       05  E1 0021B 27$:   BBC      #5, MOUNT_OPTIONS+4, 28$               ; 1176
                                 5E  DD 00220         PUSHL    SP                                     ; 1178
                             48  A6  9F 00222         PUSHAB   RECORDSZ                               ;
                                 04  DD 00225         PUSHL    #4                                     ;
                                 10  DD 00227         PUSHL    #16                                    ;
                        67       04  FB 00229         CALLS    #4, BUILD_LIST                         ;
                        0C   FB  A6  E9 0022C 28$:   BLBC     MOUNT_OPTIONS+3, 29$                   ; 1182
                                 5E  DD 00230         PUSHL    SP                                     ; 1184
                             54  A6  9F 00232         PUSHAB   WINDOW                                 ;
                                 04  DD 00235         PUSHL    #4                                     ;
                                 11  DD 00237         PUSHL    #17                                    ;
                        67       04  FB 00239         CALLS    #4, BUILD_LIST                         ;
      50   FF  A6       01       06  EF 0023C 29$:   EXTZV    #6, #1, MOUNT_OPTIONS+7, R0            ; 1190
  03  A6       01       04       50  F0 00242         INSV     R0, #4, #1, MOUNT_FLAGS+3              ;
      50   F9  A6       01       03  EF 00248         EXTZV    #3, #1, MOUNT_OPTIONS+1, R0            ; 1191
      51   F9  A6       01       04  EF 0024E         EXTZV    #4, #1, MOUNT_OPTIONS+1, R1            ;
                        50       51  88 00254         BISB2    R1, R0                                 ;
      66       01       00       50  F0 00257         INSV     R0, #0, #1, MOUNT_FLAGS                ;
      50   F8  A6       01       07  EF 0025C         EXTZV    #7, #1, MOUNT_OPTIONS, R0              ; 1192
      66       01       01       50  F0 00262         INSV     R0, #1, #1, MOUNT_FLAGS                ;
      50   FF  A6       01       02  EF 00267         EXTZV    #2, #1, MOUNT_OPTIONS+7, R0            ; 1193
  03  A6       01       00       50  F0 0026D         INSV     R0, #0, #1, MOUNT_FLAGS+3              ;
      50   FF  A6       01       03  EF 00273         EXTZV    #3, #1, MOUNT_OPTIONS+7, R0            ; 1194
  03  A6       01       01       50  F0 00279         INSV     R0, #1, #1, MOUNT_FLAGS+3              ;
      50   FE  A6       01       03  EF 0027F         EXTZV    #3, #1, MOUNT_OPTIONS+6, R0            ; 1195
  01  A6       01       05       50  F0 00285         INSV     R0, #5, #1, MOUNT_FLAGS+1              ;
  02  A6       01       06   FF  A6  F0 0028B         INSV     MOUNT_OPTIONS+7, #6, #1, MOUNT_FLAGS+2 ; 1196
      50   FE  A6       01       02  EF 00292         EXTZV    #2, #1, MOUNT_OPTIONS+6, R0            ; 1197
                        50       50  D2 00298         MCOML    R0, R0                                 ;
      66       01       02       50  F0 0029B         INSV     R0, #2, #1, MOUNT_FLAGS                ;
      50   FF  A6       01       01  EF 002A0         EXTZV    #1, #1, MOUNT_OPTIONS+7, R0            ; 1198
  02  A6       01       07       50  F0 002A6         INSV     R0, #7, #1, MOUNT_FLAGS+2              ;
      50   FE  A6       01       04  EF 002AC         EXTZV    #4, #1, MOUNT_OPTIONS+6, R0            ; 1199
  02  A6       01       01       50  F0 002B2         INSV     R0, #1, #1, MOUNT_FLAGS+2              ;
      50   FD  A6       01       02  EF 002B8         EXTZV    #2, #1, MOUNT_OPTIONS+5, R0            ; 1200
      66       01       03       50  F0 002BE         INSV     R0, #3, #1, MOUNT_FLAGS                ;
      50   FD  A6       01       04  EF 002C3         EXTZV    #4, #1, MOUNT_OPTIONS+5, R0            ; 1201
      66       01       04       50  F0 002C9         INSV     R0, #4, #1, MOUNT_FLAGS                ;
```

```
        50      FE  A6          01                07  EF 002CE        EXTZV   #7, #1, MOUNT_OPTIONS+6, R0                 ; 1202
    02  A6          01          05                50  F0 002D4        INSV    R0, #5, #1, MOUNT_FLAGS+2                   ;
        50      FE  A6          01                06  EF 002DA        EXTZV   #6, #1, MOUNT_OPTIONS+6, R0                 ; 1203
                                50                50  D2 002E0        MCOML   R0, R0                                      ;
    02  A6          01          03                50  F0 002E3        INSV    R0, #3, #1, MOUNT_FLAGS+2                   ;
        50      F9  A6          01                02  EF 002E9        EXTZV   #2, #1, MOUNT_OPTIONS+1, R0                 ; 1204
    02  A6          01          04                50  F0 002EF        INSV    R0, #4, #1, MOUNT_FLAGS+2                   ;
        50      F9  A6          01                01  EF 002F5        EXTZV   #1, #1, MOUNT_OPTIONS+1, R0                 ; 1205
                                50                50  D2 002FB        MCOML   R0, R0                                      ;
        66          01          06                50  F0 002FE        INSV    R0, #6, #1, MOUNT_FLAGS                     ;
        50      FC  A6          01                06  EF 00303        EXTZV   #6, #1, MOUNT_OPTIONS+4, R0                 ; 1206
        66          01          07                50  F0 00309        INSV    R0, #7, #1, MOUNT_FLAGS                     ;
        50      FA  A6          01                04  EF 0030E        EXTZV   #4, #1, MOUNT_OPTIONS+2, R0                 ; 1207
    01  A6          01          00                50  F0 00314        INSV    R0, #0, #1, MOUNT_FLAGS+1                   ;
        50      FA  A6          01                06  EF 0031A        EXTZV   #6, #1, MOUNT_OPTIONS+2, R0                 ; 1208
    01  A6          01          01                50  F0 00320        INSV    R0, #1, #1, MOUNT_FLAGS+1                   ;
        50      FE  A6          01                05  EF 00326        EXTZV   #5, #1, MOUNT_OPTIONS+6, R0                 ; 1209
    02  A6          01          02                50  F0 0032C        INSV    R0, #2, #1, MOUNT_FLAGS+2                   ;
        50      FA  A6          01                05  EF 00332        EXTZV   #5, #1, MOUNT_OPTIONS+2, R0                 ; 1210
    01  A6          01          02                50  F0 00338        INSV    R0, #2, #1, MOUNT_FLAGS+1                   ;
        50      FF  A6          01                04  EF 0033E        EXTZV   #4, #1, MOUNT_OPTIONS+7, R0                 ; 1211
    03  A6          01          02                50  F0 00344        INSV    R0, #2, #1, MOUNT_FLAGS+3                   ;
        50      FC  A6          01                03  EF 0034A        EXTZV   #3, #1, MOUNT_OPTIONS+4, R0                 ; 1212
    01  A6          01          03                50  F0 00350        INSV    R0, #3, #1, MOUNT_FLAGS+1                   ;
        50      F8  A6          01                06  EF 00356        EXTZV   #6, #1, MOUNT_OPTIONS, R0                   ; 1213
    01  A6          01          04                50  F0 0035C        INSV    R0, #4, #1, MOUNT_FLAGS+1                   ;
    01  A6          01          06          F9    A6  F0 00362        INSV    MOUNT_OPTIONS+1, #6, #1, MOUNT_FLAGS+1      ; 1214
        50      FC  A6          01                04  EF 00369        EXTZV   #4, #1, MOUNT_OPTIONS+4, R0                 ; 1215
    01  A6          01          07                50  F0 0036F        INSV    R0, #7, #1, MOUNT_FLAGS+1                   ;
        50      FD  A6          01                06  EF 00375        EXTZV   #6, #1, MOUNT_OPTIONS+5, R0                 ; 1216
    02  A6          01          00                50  F0 0037B        INSV    R0, #0, #1, MOUNT_FLAGS+2                   ;
        50      FF  A6          01                07  EF 00381        EXTZV   #7, #1, MOUNT_OPTIONS+7, R0                 ; 1217
    03  A6          01          05                50  F0 00387        INSV    R0, #5, #1, MOUNT_FLAGS+3                   ;
                                          4040    8F  BB 0038D        PUSHR   #^M<R6,SP>                                  ; 1222
                                                  04  DD 00391        PUSHL   #4                                          ;
                                                  04  DD 00393        PUSHL   #4                                          ;
                                67                04  FB 00395        CALLS   #4, BUILD_LIST                              ;
                                            00    BE  D4 00398        CLRL    @END_OF_LIST                                ; 1223
                                            04    AE  9F 0039B        PUSHAB  ITEM_LIST                                   ; 1229
                     00000000G  00                01  FB 0039E        CALLS   #1, SYS$MOUNT                               ;
                                                      04 003A5        RET                                                 ; 1233
                                                    0000 003A6  30$:  .WORD   Save nothing                                ; 0974
                                                  7E  D4 003A8        CLRL    -(SP)                                       ;
                                                  5E  DD 003AA        PUSHL   SP                                          ;
                                7E          04    AC  7D 003AC        MOVQ    4(AP), -(SP)                                ;
                         0000V  CF                03  FB 003B0        CALLS   #3, MAIN_HANDLER                            ;
                                                      04 003B5        RET                                                 ;

; Routine Size:  950 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 ROUTINE PARSE_QUALIFIER : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine parses the qualifiers of the MOUNT command line by
1 !       calling the CLI result parse routines.
1 !
1 ! CALLING SEQUENCE:
1 !       PARSE_QUALIFIER ()
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       MOUNT_OPTIONS BITS SET
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2
2
2 ! First, parse the qualifiers that do not have values, and cannot be negated.
2 !
2 ! /FOREIGN qualifier
2 !
2
2 IF CLISPRESENT ( FOREIGN_DESC )
2 THEN
2     MOUNT_OPTIONS [OPT_FOREIGN] = 1
2 ELSE
2     MOUNT_OPTIONS [OPT_FOREIGN] = 0;
2
2
2 ! /LABEL qualifier
2 !
2 IF CLISPRESENT ( LABEL_DESC )
2 THEN
3     BEGIN
3     MOUNT_OPTIONS [OPT_LABEL] = 1;
3     MOUNT_OPTIONS [OPT_NOLABEL] = 0;
2     END;
2
2 ! /NOLABEL qualifier
```

```
;  632          1291  2 !
;  633          1292  2 IF CLISPRESENT ( NOLABEL_DESC )
;  634          1293  2 THEN
;  635          1294  3     BEGIN
;  636          1295  3     MOUNT_OPTIONS [OPT_NOLABEL] = 1;
;  637          1296  3     MOUNT_OPTIONS [OPT_LABEL] = 0;
;  638          1297  2     END;
;  639          1298  2
;  640          1299  2
```

```
!
! Now, parse those qualifiers that do not require a value, and can be
! negated
!
! /ASSIST qualifier
!
SELECTONE CLI$PRESENT ( ASSIST_DESC ) OF
SET
    [CLI$_PRESENT,
     CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_ASSIST] = 1;
    [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_ASSIST] = 0;
TES;

! /AUTOMATIC qualifier
!
SELECTONE CLI$PRESENT ( AUTOMATIC_DESC) OF
SET
    [CLI$_PRESENT,
     CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_NOAUTO] = 0;
    [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_NOAUTO] = 1;
TES;


! /CLUSTER qualifier (default is /NOCLUSTER)
!
SELECTONE CLI$PRESENT ( CLUSTER_DESC ) OF
SET
    [CLI$_PRESENT]   :  MOUNT_OPTIONS [OPT_CLUSTER] = 1;
    [CLI$_DEFAULTED,
     CLI$_ABSENT,
     CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_CLUSTER] = 0;
TES;


! /GROUP qualifier
!
SELECTONE CLI$PRESENT ( GROUP_DESC ) OF
SET
    [CLI$_PRESENT]   :  MOUNT_OPTIONS [OPT_GROUP] = 1;
    [CLI$_DEFAULTED,
     CLI$_ABSENT,
     CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_GROUP] = 0;
TES;


! /HDR3 qualifier
!
SELECTONE CLI$PRESENT ( HDR3_DESC ) OF
SET
    [CLI$_PRESENT,
     CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_NOHDR3] = 0;
    [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_NOHDR3] = 1;
TES;

! /MESSAGE qualifier
!
SELECTONE CLI$PRESENT ( MESSAGE_DESC ) OF
```

```
2 SET
2     [CLI$_PRESENT,
2      CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_MESSAGE] = 1;
2     [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_MESSAGE] = 0;
2 TES;
2
2 ! /MOUNT_VERIFICATION qualifier
2 !
2 SELECTONE CLI$PRESENT ( MOUNT_VER_DESC ) OF
2 SET
2     [CLI$_PRESENT,
2      CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_MOUNTVER] = 1;
2     [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_MOUNTVER] = 0;
2 TES;
2
2 ! /QUOTA qualifier
2 !
2 SELECTONE CLI$PRESENT ( QUOTA_DESC ) OF
2 SET
2     [CLI$_PRESENT,
2      CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_NOQUOTA] = 0;
2     [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_NOQUOTA] = 1;
2 TES;
2
2 ! /SHARE qualifier (default is NOSHARE)
2 !
2 SELECTONE CLI$PRESENT ( SHARE_DESC ) OF
2 SET
2     [CLI$_PRESENT]   :  BEGIN
3                         MOUNT_OPTIONS [OPT_SHARE] = 1;
3                         MOUNT_OPTIONS [OPT_NOSHARE] = 0;
2                         END;
2     [CLI$_DEFAULTED,
2      CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_NOSHARE] = 1;
2 TES;
2
2
2 ! /SYSTEM qualifier
2 !
2 SELECTONE CLI$PRESENT ( SYSTEM_DESC ) OF
2 SET
2     [CLI$_PRESENT]   :  MOUNT_OPTIONS [OPT_SYSTEM] = 1;
2     [CLI$_DEFAULTED,
2      CLI$_ABSENT,
2      CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_SYSTEM] = 0;
2 TES;
2
2
2 ! /UNLOAD qualifier
2 !
2 SELECTONE CLI$PRESENT ( UNLOAD_DESC ) OF
2 SET
2     [CLI$_PRESENT,
2      CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_NOUNLOAD] = 0;
2     [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_NOUNLOAD] = 1;
2 TES;
2
```

```
;  2 ! /WRITE qualifier
;  2 !
;  2 SELECTONE CLI$PRESENT ( WRITE_DESC ) OF
;  2 SET
;  2     [CLI$_PRESENT,
;  2      CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_WRITE] = 1;
;  2     [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_WRITE] = 0;
;  2 TES;
;  2
;  2 ! /[NO]REBUILD qualifier
;  2 !
;  2 SELECTONE CLI$PRESENT ( REBUILD_DESC ) OF
;  2 SET
;  2     [CLI$_PRESENT,
;  2      CLI$_DEFAULTED] :  MOUNT_OPTIONS [OPT_NOREBUILD] = 0;
;  2     [CLI$_NEGATED]   :  MOUNT_OPTIONS [OPT_NOREBUILD] = 1;
;  2 TES;
```

```
! Finally, parse the qualifiers that might have values, or require values
!
! /ACCESSED qualifier
!
IF ( MOUNT_OPTIONS [OPT_ACCESSED] = CLI$PRESENT (ACCESSED_DESC) )
THEN
    BEGIN
    CLI$GET_VALUE ( ACCESSED_DESC, CLI_DESC );
    IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
                           .CLI_DESC [DSC$A_POINTER],
                           ACCESS ) )
    THEN
        ERR_EXIT (MOUN$_VALCNVERR);
    END;

! /BIND qualifier
!
IF ( MOUNT_OPTIONS [OPT_BIND] = CLI$PRESENT (BIND_DESC) )
THEN
    BEGIN
    CLI$GET_VALUE ( BIND_DESC, CLI_DESC );
    CH$FILL ( 0, DSC$C_S_BLN, STRUCT_NAME );
    STRUCT_NAME [DSC$B_DTYPE] = DSC$K_DTYPE_T;
    STRUCT_NAME [DSC$B_CLASS] = DSC$K_CLASS_D;
    STR$COPY_DX ( STRUCT_NAME, CLI_DESC );
    END;

! /BLOCKSIZE qualifier
!
IF ( MOUNT_OPTIONS [OPT_BLOCK] = CLI$PRESENT (BLOCK_DESC) )
THEN
    BEGIN
    CLI$GET_VALUE ( BLOCK_DESC, CLI_DESC );
    IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
                           .CLI_DESC [DSC$A_POINTER],
                           BLOCKSZ) )
    THEN
        ERR_EXIT (MOUN$_VALCNVERR);

    IF .BLOCKSZ GTRU 65534
    THEN
        ERR_EXIT (MOUN$_SZTOOBIG);
    MOUNT_OPTIONS [OPT_BLOCKSIZE] = 1;
    END;

! /CACHE qualifier.  If the /NOCACHE qualifier was explicit, then inhibit
! all options.
!
SELECTONE CLI$PRESENT (CACHE_DESC) OF
SET
    [CLI$_PRESENT] : BEGIN
                     MOUNT_OPTIONS [OPT_CACHE] = 1;
                     CACHE_ACT ();
                     END;
    [CLI$_NEGATED] : BEGIN
                     MOUNT_OPTIONS [OPT_NOCACHE] = 1;
```

```
3                               MOUNT_OPTIONS [OPT_WTHRU] = 1;
3                               MOUNT_OPTIONS [OPT_NOEXT_C] = 1;
3                               MOUNT_OPTIONS [OPT_NOFID_C] = 1;
3                               MOUNT_OPTIONS [OPT_NOQUO_C] = 1;
3                               END;
2   TES;
2
2   ! /COMMENT qualifier
2   !
3   IF ( MOUNT_OPTIONS [OPT_COMMENT] = CLI$PRESENT (COMMENT_DESC) )
2   THEN
3       BEGIN
3       CLI$GET_VALUE ( COMMENT_DESC, CLI_DESC );
3       CH$FILL ( 0, DSC$C_S_BLN, COMMENT_STRING );
3       COMMENT_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_T;
3       COMMENT_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
3       STR$COPY_DX ( COMMENT_STRING, CLI_DESC );
2       END;
2
2   ! /DATA_CHECK qualifier (value not required)
2   !
2   IF CLI$PRESENT (DATA_DESC)
2   THEN
2       DATACHECK_ACT ();
2
2   ! /DENSITY qualifier
2   !
3   IF ( MOUNT_OPTIONS [OPT_DENSITY] = CLI$PRESENT (DENSITY_DESC) )
2   THEN
2       DENSITY_ACT ();
2
2   ! /EXTENSION qualifier
2   !
3   IF ( MOUNT_OPTIONS [OPT_EXTENSION] = CLI$PRESENT (EXTENSION_DESC) )
2   THEN
3       BEGIN
3       CLI$GET_VALUE ( EXTENSION_DESC, CLI_DESC );
4       IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
4                              .CLI_DESC [DSC$A_POINTER],
4                              EXTENSION ) )
3       THEN
3           ERR_EXIT (MOUN$_VALCNVERR);
2       END;
2
2   ! /INITIALIZE qualifier
2   !
2
2   IF CLI$PRESENT ( INITIALIZE_DESC )
2   THEN
2       INITIALIZE_ACT ();
2
2   ! /JOURNAL qualifier (value not required)
2   !
2   !**JNL** SELECTONE CLI$PRESENT (JOURNAL_DESC) OF
2   !**JNL** SET
2   !**JNL**      [CLI$_PRESENT] : JOURNAL_ACT ();
2   !**JNL**      [CLI$_NEGATED] : BEGIN
```

```
2 !**JNL**                     MOUNT_OPTIONS [OPT_NOJRNL] = 1;
2 !**JNL**                     MOUNT_OPTIONS [OPT_NEWJRNL] = 0;
2 !**JNL**                     JRNL_SIZE = 0;
2 !**JNL**                     JRNL_EXTEND = 0;
2 !**JNL**                     JRNL_QUOTA = 0;
2 !**JNL**                     JRNL_RECORD_SIZE = 0;
2 !**JNL**                     END;
2 !**JNL** TES;
2
2 ! /OVERRIDE qualifier
2 !
2 IF CLI$PRESENT (OVERRIDE_DESC)
2 THEN
2     OVERRIDE_ACT ();
2
2 ! /OWNER_UIC qualifier
2 !
3 IF ( MOUNT_OPTIONS [OPT_OWNER_UIC] = CLI$PRESENT (OWNER_DESC) )
2 THEN
2     OWNER_UIC_ACT ();
2
2 ! /PROCESSOR qualifier
2 !
2 IF CLI$PRESENT (PROCESSOR_DESC)
2 THEN
2     PROCESSOR_ACT ();
2
2 ! /PROTECTION qualifier
2 !
3 IF ( MOUNT_OPTIONS [OPT_PROTECTION] = CLI$PRESENT (PROTECTION_DESC) )
2 THEN
2     PROTECTION_ACT ();
2
2 ! /RECORDSIZE qualifier
2 !
3 IF ( MOUNT_OPTIONS [OPT_RECORDSZ] = CLI$PRESENT (RECORD_DESC) )
2 THEN
3     BEGIN
3     CLI$GET_VALUE ( RECORD_DESC, CLI_DESC );
4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
4                            .CLI_DESC [DSC$A_POINTER],
4                            RECORDSZ ) )
3     THEN
3         ERR_EXIT (MOUN$_VALCNVERR);
3
3     IF .RECORDSZ GTRU 65534
3     THEN
3         ERR_EXIT (MOUN$_SZTOOBIG);
3     END;
2
2 ! /WINDOWS qualifier
2 !
3 IF ( MOUNT_OPTIONS [OPT_WINDOW] = CLI$PRESENT (WINDOW_DESC) )
2 THEN
3     BEGIN
3     CLI$GET_VALUE ( WINDOW_DESC, CLI_DESC );
```

```
;  945        1602  4      IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  946        1603  4                             .CLI_DESC [DSC$A_POINTER],
;  947        1604  4                             WINDOW ) )
;  948        1605  3      THEN
;  949        1606  3          ERR_EXIT (MOUN$_VALCNVERR);
;  950        1607  2      END;
;  951        1608  2
;  952        1609  1 END;                                        ! of PARSE_QUALIFIER routine



                                  0FFC 00000 PARSE_QUALIFIER:
                                                        .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11              ; 1234
               5B 00000000G   8F   D0 00002             MOVL     #CLI$_DEFAULTED, R11                              :
               5A 00000000G   8F   D0 00009             MOVL     #CLI$_NEGATED, R10                                :
               59 00000000G   8F   D0 00010             MOVL     #CLI$_PRESENT, R9                                 :
               58     0000'   CF   9E 00017             MOVAB    ACCESSED_DESC, R8                                 :
               57 00000000G   00   9E 0001C             MOVAB    CLI$PRESENT, R7                                   :
               56     0000'   CF   9E 00023             MOVAB    MOUNT_OPTIONS, R6                                 :
                      00BC    C8   9F 00028             PUSHAB   FOREIGN_DESC                                      ; 1274
               67             01   FB 0002C             CALLS    #1, CLI$PRESENT                                   :
               06             50   E9 0002F             BLBC     R0, 1$                                            :
         01    A6             08   88 00032             BISB2    #8, MOUNT_OPTIONS+1                               ; 1276
                              04   11 00036             BRB      2$                                                :
         01    A6             08   8A 00038 1$:         BICB2    #8, MOUNT_OPTIONS+1                               ; 1278
                      010C    C8   9F 0003C 2$:         PUSHAB   LABEL_DESC                                        ; 1283
               67             01   FB 00040             CALLS    #1, CLI$PRESENT                                   :
               09             50   E9 00043             BLBC     R0, 3$                                            :
         03    A6        80   8F   88 00046             BISB2    #128, MOUNT_OPTIONS+3                             ; 1286
         01    A6             10   8A 0004B             BICB2    #16, MOUNT_OPTIONS+1                              ; 1287
                      0148    C8   9F 0004F 3$:         PUSHAB   NOLABEL_DESC                                      ; 1292
               67             01   FB 00053             CALLS    #1, CLI$PRESENT                                   :
               09             50   E9 00056             BLBC     R0, 4$                                            :
         01    A6             10   88 00059             BISB2    #16, MOUNT_OPTIONS+1                              ; 1295
         03    A6        80   8F   8A 0005D             BICB2    #128, MOUNT_OPTIONS+3                             ; 1296
                         10   A8   9F 00062 4$:         PUSHAB   ASSIST_DESC                                       ; 1306
               67             01   FB 00065             CALLS    #1, CLI$PRESENT                                   :
               59             50   D1 00068             CMPL     R0, R9                                            ; 1308
                              05   13 0006B             BEQL     5$                                                :
               5B             50   D1 0006D             CMPL     R0, R11                                           :
                              06   12 00070             BNEQ     6$                                                :
         06    A6             04   88 00072 5$:         BISB2    #4, MOUNT_OPTIONS+6                               ; 1309
                              09   11 00076             BRB      7$                                                :
               5A             50   D1 00078 6$:         CMPL     R0, R10                                           ; 1310
                              04   12 0007B             BNEQ     7$                                                :
         06    A6             04   8A 0007D             BICB2    #4, MOUNT_OPTIONS+6                               :
                         24   A8   9F 00081 7$:         PUSHAB   AUTOMATIC_DESC                                    ; 1315
               67             01   FB 00084             CALLS    #1, CLI$PRESENT                                   :
               59             50   D1 00087             CMPL     R0, R9                                            ; 1317
                              05   13 0008A             BEQL     8$                                                :
               5B             50   D1 0008C             CMPL     R0, R11                                           :
                              06   12 0008F             BNEQ     9$                                                :
         07    A6             02   8A 00091 8$:         BICB2    #2, MOUNT_OPTIONS+7                               ; 1318
                              09   11 00095             BRB      10$                                               :
               5A             50   D1 00097 9$:         CMPL     R0, R10                                           ; 1319
```

```
                                 04   12  0009A          BNEQ     10$                                    ;
             07   A6             02   88  0009C          BISB2    #2, MOUNT_OPTIONS+7                    ;
                           64    A8   9F  000A0   10$:   PUSHAB   CLUSTER_DESC                           ; 1325
                  67             01   FB  000A3          CALLS    #1, CLI$PRESENT                        ;
                  59             50   D1  000A6          CMPL     R0, R9                                 ; 1327
                                 07   12  000A9          BNEQ     11$                                    ;
             07   A6       40    8F   88  000AB          BISB2    #64, MOUNT_OPTIONS+7                   ;
                                 18   11  000B0          BRB      13$                                    ;
                  5B             50   D1  000B2   11$:   CMPL     R0, R11                                ; 1328
                                 0E   13  000B5          BEQL     12$                                    ;
      00000000G   8F             50   D1  000B7          CMPL     R0, #CLI$_ABSENT                       ;
                                 05   13  000BE          BEQL     12$                                    ;
                  5A             50   D1  000C0          CMPL     R0, R10                                ;
                                 05   12  000C3          BNEQ     13$                                    ;
             07   A6       40    8F   8A  000C5   12$:   BICB2    #64, MOUNT_OPTIONS+7                   ; 1330
                         00CC    C8   9F  000CA   13$:   PUSHAB   GROUP_DESC                             ; 1336
                  67             01   FB  000CE          CALLS    #1, CLI$PRESENT                        ;
                  59             50   D1  000D1          CMPL     R0, R9                                 ; 1338
                                 06   12  000D4          BNEQ     14$                                    ;
                  66       80    8F   88  000D6          BISB2    #128, MOUNT_OPTIONS                    ;
                                 17   11  000DA          BRB      16$                                    ;
                  5B             50   D1  000DC   14$:   CMPL     R0, R11                                ; 1339
                                 0E   13  000DF          BEQL     15$                                    ;
      00000000G   8F             50   D1  000E1          CMPL     R0, #CLI$_ABSENT                       ;
                                 05   13  000E8          BEQL     15$                                    ;
                  5A             50   D1  000EA          CMPL     R0, R10                                ;
                                 04   12  000ED          BNEQ     16$                                    ;
                  66       80    8F   8A  000EF   15$:   BICB2    #128, MOUNT_OPTIONS                    ; 1341
                         00D8    C8   9F  000F3   16$:   PUSHAB   HDR3_DESC                              ; 1347
                  67             01   FB  000F7          CALLS    #1, CLI$PRESENT                        ;
                  59             50   D1  000FA          CMPL     R0, R9                                 ; 1349
                                 05   13  000FD          BEQL     17$                                    ;
                  5B             50   D1  000FF          CMPL     R0, R11                                ;
                                 06   12  00102          BNEQ     18$                                    ;
             05   A6             10   8A  00104   17$:   BICB2    #16, MOUNT_OPTIONS+5                   ; 1350
                                 09   11  00108          BRB      19$                                    ;
                  5A             50   D1  0010A   18$:   CMPL     R0, R10                                ; 1351
                                 04   12  0010D          BNEQ     19$                                    ;
             05   A6             10   88  0010F          BISB2    #16, MOUNT_OPTIONS+5                   ;
                         011C    C8   9F  00113   19$:   PUSHAB   MESSAGE_DESC                           ; 1356
                  67             01   FB  00117          CALLS    #1, CLI$PRESENT                        ;
                  59             50   D1  0011A          CMPL     R0, R9                                 ; 1358
                                 05   13  0011D          BEQL     20$                                    ;
                  5B             50   D1  0011F          CMPL     R0, R11                                ;
                                 06   12  00122          BNEQ     21$                                    ;
             06   A6             08   88  00124   20$:   BISB2    #8, MOUNT_OPTIONS+6                    ; 1359
                                 09   11  00128          BRB      22$                                    ;
                  5A             50   D1  0012A   21$:   CMPL     R0, R10                                ; 1360
                                 04   12  0012D          BNEQ     22$                                    ;
             06   A6             08   8A  0012F          BICB2    #8, MOUNT_OPTIONS+6                    ;
                         0138    C8   9F  00133   22$:   PUSHAB   MOUNT_VER_DESC                         ; 1365
                  67             01   FB  00137          CALLS    #1, CLI$PRESENT                        ;
                  59             50   D1  0013A          CMPL     R0, R9                                 ; 1367
                                 05   13  0013D          BEQL     23$                                    ;
                  5B             50   D1  0013F          CMPL     R0, R11                                ;
                                 07   12  00142          BNEQ     24$                                    ;
             06   A6       40    8F   88  00144   23$:   BISB2    #64, MOUNT_OPTIONS+6                   ; 1368
```

```
                        0A  11  00149         BRB     25$                          ;
            5A          50  D1  0014B 24$:    CMPL    R0, R10                      ; 1369
                        05  12  0014E         BNEQ    25$                          ;
       06   A6      40  8F  8A  00150         BICB2   #64, MOUNT_OPTIONS+6         ;
                  01A4  C8  9F  00155 25$:    PUSHAB  QUOTA_DESC                   ; 1374
            67          01  FB  00159         CALLS   #1, CLI$PRESENT              ;
            59          50  D1  0015C         CMPL    R0, R9                       ; 1376
                        05  13  0015F         BEQL    26$                          ;
            5B          50  D1  00161         CMPL    R0, R11                      ;
                        06  12  00164         BNEQ    27$                          ;
       05   A6          04  8A  00166 26$:    BICB2   #4, MOUNT_OPTIONS+5          ; 1377
                        09  11  0016A         BRB     28$                          ;
            5A          50  D1  0016C 27$:    CMPL    R0, R10                      ; 1378
                        04  12  0016F         BNEQ    28$                          ;
       05   A6          04  88  00171         BISB2   #4, MOUNT_OPTIONS+5          ;
                  01D8  C8  9F  00175 28$:    PUSHAB  SHARE_DESC                   ; 1383
            67          01  FB  00179         CALLS   #1, CLI$PRESENT              ;
            59          50  D1  0017C         CMPL    R0, R9                       ; 1385
                        09  12  0017F         BNEQ    29$                          ;
            66      40  8F  88  00181         BISB2   #64, MOUNT_OPTIONS           ; 1386
            66          10  8A  00185         BICB2   #16, MOUNT_OPTIONS           ; 1387
                        0D  11  00188         BRB     31$                          ; 1383
            5B          50  D1  0018A 29$:    CMPL    R0, R11                      ; 1389
                        05  13  0018D         BEQL    30$                          ;
            5A          50  D1  0018F         CMPL    R0, R10                      ;
                        03  12  00192         BNEQ    31$                          ;
            66          10  88  00194 30$:    BISB2   #16, MOUNT_OPTIONS           ; 1390
                  01E8  C8  9F  00197 31$:    PUSHAB  SYSTEM_DESC                  ; 1396
            67          01  FB  0019B         CALLS   #1, CLI$PRESENT              ;
            59          50  D1  0019E         CMPL    R0, R9                       ; 1398
                        06  12  001A1         BNEQ    32$                          ;
       01   A6          01  88  001A3         BISB2   #1, MOUNT_OPTIONS+1          ;
                        17  11  001A7         BRB     34$                          ;
            5B          50  D1  001A9 32$:    CMPL    R0, R11                      ; 1399
                        0E  13  001AC         BEQL    33$                          ;
00000000G   8F          50  D1  001AE         CMPL    R0, #CLI$_ABSENT             ;
                        05  13  001B5         BEQL    33$                          ;
            5A          50  D1  001B7         CMPL    R0, R10                      ;
                        04  12  001BA         BNEQ    34$                          ;
       01   A6          01  8A  001BC 33$:    BICB2   #1, MOUNT_OPTIONS+1          ; 1401
                  01F8  C8  9F  001C0 34$:    PUSHAB  UNLOAD_DESC                  ; 1407
            67          01  FB  001C4         CALLS   #1, CLI$PRESENT              ;
            59          50  D1  001C7         CMPL    R0, R9                       ; 1409
                        05  13  001CA         BEQL    35$                          ;
            5B          50  D1  001CC         CMPL    R0, R11                      ;
                        06  12  001CF         BNEQ    36$                          ;
       01   A6          04  8A  001D1 35$:    BICB2   #4, MOUNT_OPTIONS+1          ; 1410
                        09  11  001D5         BRB     37$                          ;
            5A          50  D1  001D7 36$:    CMPL    R0, R10                      ; 1411
                        04  12  001DA         BNEQ    37$                          ;
       01   A6          04  88  001DC         BISB2   #4, MOUNT_OPTIONS+1          ;
                  0218  C8  9F  001E0 37$:    PUSHAB  WRITE_DESC                   ; 1416
            67          01  FB  001E4         CALLS   #1, CLI$PRESENT              ;
            59          50  D1  001E7         CMPL    R0, R9                       ; 1418
                        05  13  001EA         BEQL    38$                          ;
            5B          50  D1  001EC         CMPL    R0, R11                      ;
                        06  12  001EF         BNEQ    39$                          ;
```

```
                           01  A6            02  88  001F1 38$:   BISB2    #2, MOUNT_OPTIONS+1                          : 1419
                                             09  11  001F5        BRB      40$                                          :
                               5A            50  D1  001F7 39$:   CMPL     R0, R10                                      : 1420
                                             04  12  001FA        BNEQ     40$                                          :
                           01  A6            02  8A  001FC        BICB2    #2, MOUNT_OPTIONS+1                          :
                                       01B4  C8  9F  00200 40$:   PUSHAB   REBUILD_DESC                                 : 1425
                               67            01  FB  00204        CALLS    #1, CLI$PRESENT                              :
                               59            50  D1  00207        CMPL     R0, R9                                       : 1427
                                             05  13  0020A        BEQL     41$                                          :
                               5B            50  D1  0020C        CMPL     R0, R11                                      :
                                             07  12  0020F        BNEQ     42$                                          :
                           07  A6        80  8F  8A  00211 41$:   BICB2    #128, MOUNT_OPTIONS+7                        : 1428
                                             0A  11  00216        BRB      43$                                          :
                               5A            50  D1  00218 42$:   CMPL     R0, R10                                      : 1429
                                             05  12  0021B        BNEQ     43$                                          :
                           07  A6        80  8F  88  0021D        BISB2    #128, MOUNT_OPTIONS+7                        :
                                             58  DD  00222 43$:   PUSHL    R8                                           : 1436
                               67            01  FB  00224        CALLS    #1, CLI$PRESENT                              :
03   A6         01             01            50  F0  00227        INSV     R0, #1, #1, MOUNT_OPTIONS+3                  :
                               2D            50  E9  0022D        BLBC     R0, 44$                                      :
                                         60  A6  9F  00230        PUSHAB   CLI_DESC                                     : 1439
                                             58  DD  00233        PUSHL    R8                                           :
                    00000000G  00            02  FB  00235        CALLS    #2, CLI$GET_VALUE                            :
                                         0C  A6  9F  0023C        PUSHAB   ACCESS                                       : 1440
                                         64  A6  DD  0023F        PUSHL    CLI_DESC+4                                   : 1441
                               7E        60  A6  3C  00242        MOVZWL   CLI_DESC, -(SP)                              : 1440
                    00000000G  00            03  FB  00246        CALLS    #3, LIB$CVT_DTB                              :
                               0D            50  E8  0024D        BLBS     R0, 44$                                      :
                                   0072805C  8F  DD  00250        PUSHL    #7503964                                     : 1444
                    00000000G  00            01  FB  00256        CALLS    #1, LIB$STOP                                 :
                                         30  A8  9F  0025D 44$:   PUSHAB   BIND_DESC                                    : 1449
                               67            01  FB  00260        CALLS    #1, CLI$PRESENT                              :
05   A6         01             00            50  F0  00263        INSV     R0, #0, #1, MOUNT_OPTIONS+5                  :
                               27            50  E9  00269        BLBC     R0, 45$                                      :
                                         60  A6  9F  0026C        PUSHAB   CLI_DESC                                     : 1452
                                         30  A8  9F  0026F        PUSHAB   BIND_DESC                                    :
                    00000000G  00            02  FB  00272        CALLS    #2, CLI$GET_VALUE                            :
     08         00             6E            00  2C  00279        MOVC5    #0, (SP), #0, #8, STRUCT_NAME                : 1453
                                         54  A6      0027E                                                              :
                           56  A6      020E  8F  B0  00280        MOVW     #526, STRUCT_NAME+2                          : 1454
                                         60  A6  9F  00286        PUSHAB   CLI_DESC                                     : 1456
                                         54  A6  9F  00289        PUSHAB   STRUCT_NAME                                  :
                    00000000G  00            02  FB  0028C        CALLS    #2, STR$COPY_DX                              :
                                         44  A8  9F  00293 45$:   PUSHAB   BLOCK_DESC                                   : 1461
                               67            01  FB  00296        CALLS    #1, CLI$PRESENT                              :
01   A6         01             07            50  F0  00299        INSV     R0, #7, #1, MOUNT_OPTIONS+1                  :
                               49            50  E9  0029F        BLBC     R0, 48$                                      :
                                         60  A6  9F  002A2        PUSHAB   CLI_DESC                                     : 1464
                                         44  A8  9F  002A5        PUSHAB   BLOCK_DESC                                   :
                    00000000G  00            02  FB  002A8        CALLS    #2, CLI$GET_VALUE                            :
                                         18  A6  9F  002AF        PUSHAB   BLOCKSZ                                      : 1465
                                         64  A6  DD  002B2        PUSHL    CLI_DESC+4                                   : 1466
                               7E        60  A6  3C  002B5        MOVZWL   CLI_DESC, -(SP)                              : 1465
                    00000000G  00            03  FB  002B9        CALLS    #3, LIB$CVT_DTB                              :
                               0D            50  E8  002C0        BLBS     R0, 46$                                      :
                                   0072805C  8F  DD  002C3        PUSHL    #7503964                                     : 1469
                    00000000G  00            01  FB  002C9        CALLS    #1, LIB$STOP                                 :
```

```
                    0000FFFE  8F        18  A6  D1 002D0 46$:  CMPL     BLOCKSZ, #65534                        ; 1471
                                            0D  1B 002D8       BLEQU    47$
                                  0072817C  8F  DD 002DA       PUSHL    #7504252                               ; 1473
                    00000000G 00            01  FB 002E0       CALLS    #1, LIB$STOP
                           02 A6            01  88 002E7 47$:  BISB2    #1, MOUNT_OPTIONS+2                    ; 1474
                                        54  A8  9F 002EB 48$:  PUSHAB   CACHE_DESC                             ; 1480
                              67            01  FB 002EE       CALLS    #1, CLI$PRESENT
                              59            50  D1 002F1       CMPL     R0, R9                                 ; 1482
                                            0B  12 002F4       BNEQ     49$
                           05 A6            20  88 002F6       BISB2    #32, MOUNT_OPTIONS+5                   ; 1483
                        0000V CF            00  FB 002FA       CALLS    #0, CACHE_ACT                          ; 1484
                                            0B  11 002FF       BRB      50$                                    ; 1480
                              5A            50  D1 00301 49$:  CMPL     R0, R10                                ; 1486
                                            06  12 00304       BNEQ     50$
                           05 A6      13C0  8F  A8 00306       BISW2    #5056, MOUNT_OPTIONS+5                 ; 1491
                                        74  A8  9F 0030C 50$:  PUSHAB   COMMENT_DESC                           ; 1497
                              67            01  FB 0030F       CALLS    #1, CLI$PRESENT
       66          01         03            50  F0 00312       INSV     R0, #3, #1, MOUNT_OPTIONS
                              27            50  E9 00317       BLBC     R0, 51$
                                        60  A6  9F 0031A       PUSHAB   CLI_DESC                               ; 1500
                                        74  A8  9F 0031D       PUSHAB   COMMENT_DESC
                    00000000G 00            02  FB 00320       CALLS    #2, CLI$GET_VALUE
       08          00         6E            00  2C 00327       MOVC5    #0, (SP), #0, #8, COMMENT_STRING       ; 1501
                                        28  A6     0032C
                           2A A6      020E  8F  B0 0032E       MOVW     #526, COMMENT_STRING+2                 ; 1502
                                        60  A6  9F 00334       PUSHAB   CLI_DESC                               ; 1504
                                        28  A6  9F 00337       PUSHAB   COMMENT_STRING
                    00000000G 00            02  FB 0033A       CALLS    #2, STR$COPY_DX
                                      0088  C8  9F 00341 51$:  PUSHAB   DATA_DESC                              ; 1509
                              67            01  FB 00345       CALLS    #1, CLI$PRESENT
                              05            50  E9 00348       BLBC     R0, 52$
                        0000V CF            00  FB 0034B       CALLS    #0, DATACHECK_ACT                      ; 1511
                                      0098  C8  9F 00350 52$:  PUSHAB   DENSITY_DESC                           ; 1515
                              67            01  FB 00354       CALLS    #1, CLI$PRESENT
       66          01         00            50  F0 00357       INSV     R0, #0, #1, MOUNT_OPTIONS
                              05            50  E9 0035C       BLBC     R0, 53$
                        0000V CF            00  FB 0035F       CALLS    #0, DENSITY_ACT                        ; 1517
                                      00AC  C8  9F 00364 53$:  PUSHAB   EXTENSION_DESC                         ; 1521
                              67            01  FB 00368       CALLS    #1, CLI$PRESENT
    02 A6          01         07            50  F0 0036B       INSV     R0, #7, #1, MOUNT_OPTIONS+2
                              2F            50  E9 00371       BLBC     R0, 54$
                                        60  A6  9F 00374       PUSHAB   CLI_DESC                               ; 1524
                                      00AC  C8  9F 00377       PUSHAB   EXTENSION_DESC
                    00000000G 00            02  FB 0037B       CALLS    #2, CLI$GET_VALUE
                                        34  A6  9F 00382       PUSHAB   EXTENSION                              ; 1525
                                        64  A6  DD 00385       PUSHL    CLI_DESC+4                             ; 1526
                              7E        60  A6  3C 00388       MOVZWL   CLI_DESC, -(SP)                        ; 1525
                    00000000G 00            03  FB 0038C       CALLS    #3, LIB$CVT_DTB
                              0D            50  E8 00393       BLBS     R0, 54$
                                  0072805C  8F  DD 00396       PUSHL    #7503964                               ; 1529
                    00000000G 00            01  FB 0039C       CALLS    #1, LIB$STOP
                                      00EC  C8  9F 003A3 54$:  PUSHAB   INITIALIZE_DESC                        ; 1535
                              67            01  FB 003A7       CALLS    #1, CLI$PRESENT
                              05            50  E9 003AA       BLBC     R0, 55$
                        0000V CF            00  FB 003AD       CALLS    #0, INITIALIZE_ACT                     ; 1537
                                      0158  C8  9F 003B2 55$:  PUSHAB   OVERRIDE_DESC                          ; 1556
                              67            01  FB 003B6       CALLS    #1, CLI$PRESENT
```

```
                                       05           50  E9 003B9         BLBC    R0, 56$                             ;
                            0000V      CF           00  FB 003BC         CALLS   #0, OVERRIDE_ACT                    ; 1558
                                             016C   C8  9F 003C1 56$:    PUSHAB  OWNER_DESC                          ; 1562
                                       67           01  FB 003C5         CALLS   #1, CLI$PRESENT                     ;
      02    A6            01           02           50  F0 003C8         INSV    R0, #2, #1, MOUNT_OPTIONS+2         ;
                                       05           50  E9 003CE         BLBC    R0, 57$                             ;
                            0000V      CF           00  FB 003D1         CALLS   #0, OWNER_UIC_ACT                   ; 1564
                                             0180   C8  9F 003D6 57$:    PUSHAB  PROCESSOR_DESC                      ; 1568
                                       67           01  FB 003DA         CALLS   #1, CLI$PRESENT                     ;
                                       05           50  E9 003DD         BLBC    R0, 58$                             ;
                            0000V      CF           00  FB 003E0         CALLS   #0, PROCESSOR_ACT                   ; 1570
                                             0194   C8  9F 003E5 58$:    PUSHAB  PROTECTION_DESC                     ; 1574
                                       67           01  FB 003E9         CALLS   #1, CLI$PRESENT                     ;
      02    A6            01           01           50  F0 003EC         INSV    R0, #1, #1, MOUNT_OPTIONS+2         ;
                                       05           50  E9 003F2         BLBC    R0, 59$                             ;
                            0000V      CF           00  FB 003F5         CALLS   #0, PROTECTION_ACT                  ; 1576
                                             01C8   C8  9F 003FA 59$:    PUSHAB  RECORD_DESC                         ; 1580
                                       67           01  FB 003FE         CALLS   #1, CLI$PRESENT                     ;
      04    A6            01           05           50  F0 00401         INSV    R0, #5, #1, MOUNT_OPTIONS+4         ;
                                       46           50  E9 00407         BLBC    R0, 61$                             ;
                                               60   A6  9F 0040A         PUSHAB  CLI_DESC                            ; 1583
                                             01C8   C8  9F 0040D         PUSHAB  RECORD_DESC                         ;
                        00000000G      00           02  FB 00411         CALLS   #2, CLI$GET_VALUE                   ;
                                               50   A6  9F 00418         PUSHAB  RECORDSZ                            ; 1584
                                               64   A6  DD 0041B         PUSHL   CLI_DESC+4                          ; 1585
                                       7E      60   A6  3C 0041E         MOVZWL  CLI_DESC, -(SP)                     ; 1584
                        00000000G      00           03  FB 00422         CALLS   #3, LIB$CVT_DTB                     ;
                                       0D           50  E8 00429         BLBS    R0, 60$                             ;
                                         0072805C   8F  DD 0042C         PUSHL   #7503964                            ; 1588
                        00000000G      00           01  FB 00432         CALLS   #1, LIB$STOP                        ;
                        0000FFFE       8F      50   A6  D1 00439 60$:    CMPL    RECORDSZ, #65534                    ; 1590
                                       0D           1B     00441         BLEQU   61$                                 ;
                                         0072817C   8F  DD 00443         PUSHL   #7504252                            ; 1592
                        00000000G      00           01  FB 00449         CALLS   #1, LIB$STOP                        ;
                                             0208   C8  9F 00450 61$:    PUSHAB  WINDOW_DESC                         ; 1597
                                       67           01  FB 00454         CALLS   #1, CLI$PRESENT                     ;
      03    A6            01           00           50  F0 00457         INSV    R0, #0, #1, MOUNT_OPTIONS+3         ;
                                       2F           50  E9 0045D         BLBC    R0, 62$                             ;
                                               60   A6  9F 00460         PUSHAB  CLI_DESC                            ; 1600
                                             0208   C8  9F 00463         PUSHAB  WINDOW_DESC                         ;
                        00000000G      00           02  FB 00467         CALLS   #2, CLI$GET_VALUE                   ;
                                               5C   A6  9F 0046E         PUSHAB  WINDOW                              ; 1602
                                               64   A6  DD 00471         PUSHL   CLI_DESC+4                          ; 1603
                                       7E      60   A6  3C 00474         MOVZWL  CLI_DESC, -(SP)                     ; 1602
                        00000000G      00           03  FB 00478         CALLS   #3, LIB$CVT_DTB                     ;
                                       0D           50  E8 0047F         BLBS    R0, 62$                             ;
                                         0072805C   8F  DD 00482         PUSHL   #7503964                            ; 1606
                        00000000G      00           01  FB 00488         CALLS   #1, LIB$STOP                        ;
                                                        04 0048F 62$:    RET                                         ; 1609

; Routine Size:  1168 bytes,    Routine Base:  $CODE$ + 03B6
```

```
1 ROUTINE BUILD_LIST (ITEM_CODE, ITEM_LENGTH, ITEM_ADDRESS, LIST_PTR) : NOVALUE =
1
1 !++
1 ! Functional description:
1 !
1 !       This routine will build an item list entry from the input parameters.
1 !
1 ! Input:
1 !
1 !       ITEM_ADDRESS     : Address of item
1 !       ITEM_CODE        : Item code value
1 !       ITEM_LENGTH      : Length of item (in bytes)
1 !       LIST_PTR         : Address of a pointer to the end of the list
1 !
1 !
1 ! Implicit Input:
1 !
1 !       The list is assumed to be long enough.
1 !
1 ! Output:
1 !
1 !       LIST             : points to new end of list
1 !
1 ! Implict output:
1 !
1 !       None.
1 !
1 ! Side effects:
1 !
1 !       None.
1 !
1 ! Routine value:
1 !
1 !       None.
1 !--
1
2 BEGIN                                                  ! Start of BUILD_ENTRY
2
2 LOCAL
2       LIST             : REF BBLOCK;
2
2 MACRO
2       LENGTH           =  0, 0, 16, 0%,
2       CODE             =  2, 0, 16, 0%,
2       ADDRESS          =  4, 0, 32, 0%,
2       UNUSED           =  8, 0, 32, 0%;
2
2 LIST = ..LIST_PTR;                                     ! Get address of start of entry
2 LIST [LENGTH] = .ITEM_LENGTH;                          ! Set the item length
2 LIST [CODE] = .ITEM_CODE;                              ! Set the item code
2 LIST [ADDRESS] = .ITEM_ADDRESS;                        ! Set the item address
2 LIST [UNUSED] = 0;                                     ! Clear the unused portion
2 .LIST_PTR = .LIST + ITEM_SIZE;                         ! Set new end of list
2
1 END;                                                   ! End of BUILD_ENTRY
```

```
                                   0000 00000 BUILD_LIST:
                                                   .WORD   Save nothing                              ; 1610
          50        10    BC  D0 00002             MOVL    @LIST_PTR, LIST                           ; 1657
          60        08    AC  B0 00006             MOVW    ITEM_LENGTH, (LIST)                       ; 1658
     02   A0        04    AC  B0 0000A             MOVW    ITEM_CODE, 2(LIST)                        ; 1659
     04   A0        0C    AC  D0 0000F             MOVL    ITEM_ADDRESS, 4(LIST)                     ; 1660
                    08    A0  D4 00014             CLRL    8(LIST)                                   ; 1661
     10   BC        0C    A0  9E 00017             MOVAB   12(R0), @LIST_PTR                         ; 1662
                              04 0001C             RET                                               ; 1664

; Routine Size:  29 bytes,    Routine Base:  $CODE$ + 0846
; 1010          1665  1 ROUTINE MAIN_HANDLER (SIGNAL, MECHANISM) =
; 1011          1666  1
; 1012          1667  1 !++
; 1013          1668  1 !
; 1014          1669  1 ! FUNCTIONAL DESCRIPTION:
; 1015          1670  1 !
; 1016          1671  1 !       This routine is the main level condition handler for the MOUNT
; 1017          1672  1 !       utility. It undoes anything that MOUNT has done so far and returns
; 1018          1673  1 !       the condition code as status to MOUNT's caller (i.e., the CLI).
; 1019          1674  1 !
; 1020          1675  1 !
; 1021          1676  1 ! CALLING SEQUENCE:
; 1022          1677  1 !       MAIN_HANDLER (ARG1, ARG2)
; 1023          1678  1 !
; 1024          1679  1 ! INPUT PARAMETERS:
; 1025          1680  1 !       ARG1: address of signal array
; 1026          1681  1 !       ARG2: address of mechanism array
; 1027          1682  1 !
; 1028          1683  1 ! IMPLICIT INPUTS:
; 1029          1684  1 !       NONE
; 1030          1685  1 !
; 1031          1686  1 ! OUTPUT PARAMETERS:
; 1032          1687  1 !       NONE
; 1033          1688  1 !
; 1034          1689  1 ! IMPLICIT OUTPUTS:
; 1035          1690  1 !       NONE
; 1036          1691  1 !
; 1037          1692  1 ! ROUTINE VALUE:
; 1038          1693  1 !       NONE
; 1039          1694  1 !
; 1040          1695  1 ! SIDE EFFECTS:
; 1041          1696  1 !       stack unwound, control passed to CLI
; 1042          1697  1 !
; 1043          1698  1 !--
; 1044          1699  1
; 1045          1700  2 BEGIN
; 1046          1701  2
; 1047          1702  2 MAP
; 1048          1703  2         SIGNAL          : REF BBLOCK,   ! signal array
; 1049          1704  2         MECHANISM       : REF BBLOCK;   ! mechanism array
; 1050          1705  2
; 1051          1706  2
; 1052          1707  2 ! Force the facility code to be mount and resignal the
; 1053          1708  2 ! error to be printed by the catch all handler.
; 1054          1709  2 !
; 1055          1710  2
; 1056          1711  2 IF .BBLOCK [SIGNAL[CHF$L_SIG_NAME], STS$V_FAC_NO] EQL 0
; 1057          1712  2 OR .BBLOCK [SIGNAL[CHF$L_SIG_NAME], STS$V_FAC_NO] EQL INIT$_FACILITY
; 1058          1713  2 THEN BBLOCK [SIGNAL[CHF$L_SIG_NAME], STS$V_FAC_NO] = MOUN$_FACILITY;
; 1059          1714  2
; 1060          1715  2 RETURN SS$_RESIGNAL;
; 1061          1716  2
; 1062          1717  1 END;                                    ! end of routine MAIN_HANDLER
```

```
                                                       0000 00000 MAIN_HANDLER:
                                                                          .WORD   Save nothing                                ; 1665
                                       50       04    AC  D0 00002         MOVL    SIGNAL, R0                                  ; 1711
                                0FFF   8F       06    A0  B3 00006         BITW    6(R0), #4095                                ;
                                                     0C  13 0000C         BEQL    1$                                          ;
00000075   8F       06   A0            0C            00  ED 0000E         CMPZV   #0, #12, 6(R0), #117                        ; 1712
                                                     0A  12 00018         BNEQ    2$                                          ;
      06   A0            0C            00 00000072   8F  F0 0001A 1$:     INSV    #114, #0, #12, 6(R0)                        ; 1713
                                       50    0918    8F  3C 00024 2$:     MOVZWL  #2328, R0                                   ; 1715
                                                         04 00029         RET                                                 ; 1717

; Routine Size:  42 bytes,    Routine Base:  $CODE$ + 0863
```

```
1 !+
1 !
1 ! Parameter and qualifier action routines. Each routine is named corresponding
1 ! to its associated parameter of qualifier. Each routine does whatever
1 ! conversion is necessary and stores the parameter or qualifier value in
1 ! the appropriate location in the output area.
1 !
1 !-
1
1
1 ROUTINE CACHE_ACT : NOVALUE =
2 BEGIN
2
2 EXTERNAL
2         CACHE_STB       : VECTOR [0],   ! state table address
2         CACHE_KTB       : VECTOR [0];   ! keyword table address
2
2 EXTERNAL ROUTINE
2         LIB$TPARSE;
2
2 !
2 ! Initialize work area.
2 !
2
2 EXT_CACHE = -1;                                 ! Set value for EXTENT not seen
2 FID_CACHE = -1;                                 ! Set value for FILE_ID not seen
2 QUO_CACHE = -1;                                 ! Set value for QUOTA not seen
2
2 ! Parse the cache control options and set appropriate flags.
2 !
2
2 WHILE CLI$GET_VALUE ( CACHE_DESC, CLI_DESC ) DO
3 BEGIN
3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
3     IF NOT LIB$TPARSE (TPARSE_BLOCK, CACHE_STB, CACHE_KTB)
3     THEN
3         ERR_EXIT (MOUN$_BADCACHE);
2 END;
2
2 !
2 ! Check to see if caching should be turned off:
2 !
2 !   /CACHE=EXTENT:0      disables extent caching
2 !   /CACHE=FILE_ID:1     disables FID caching
2 !   /CACHE=QUOTA:0       disables quota caching
2 !
2
2 IF .EXT_CACHE EQL 0                             ! /CACHE=EXTENT:0
2 THEN
2     MOUNT_OPTIONS [OPT_NOEXT_C] = 1;
2
2 IF .FID_CACHE EQL 1                             ! /CACHE=FILE_ID:1
2 THEN
2     MOUNT_OPTIONS [OPT_NOFID_C] = 1;
2
2 IF .QUO_CACHE EQL 0                             ! /CACHE=QUOTA:0
```

```
; 1121          1775  2 THEN
; 1122          1776  2     MOUNT_OPTIONS [OPT_NOQUO_C] = 1;
; 1123          1777  2
; 1124          1778  2
; 1125          1779  1 END;                                    ! end of routine CACHE_ACT


                                                                  .EXTRN  CACHE_STB, CACHE_KTB
                                                                  .EXTRN  LIB$TPARSE

                                          0004 00000 CACHE_ACT:
                                                                  .WORD   Save R2                                        ; 1728
                      52      0000'   CF   9E 00002               MOVAB   EXT_CACHE, R2                                  ;
                      62              01   CE 00007               MNEGL   #1, EXT_CACHE                                  ; 1742
                04    A2              01   CE 0000A               MNEGL   #1, FID_CACHE                                  ; 1743
                08    A2              01   CE 0000E               MNEGL   #1, QUO_CACHE                                  ; 1744
                                44    A2   9F 00012 1$:           PUSHAB  CLI_DESC                                       ; 1749
                              0000'   CF   9F 00015               PUSHAB  CACHE_DESC                                     ;
          00000000G   00              02   FB 00019               CALLS   #2, CLI$GET_VALUE                              ;
                      32              50   E9 00020               BLBC    R0, 2$                                         ;
                58    A2        44    A2   3C 00023               MOVZWL  CLI_DESC, TPARSE_BLOCK+8                       ; 1751
                5C    A2        48    A2   D0 00028               MOVL    CLI_DESC+4, TPARSE_BLOCK+12                    ; 1752
                         00000000G    00   9F 0002D               PUSHAB  CACHE_KTB                                      ; 1753
                         00000000G    00   9F 00033               PUSHAB  CACHE_STB                                      ;
                                50    A2   9F 00039               PUSHAB  TPARSE_BLOCK                                   ;
          00000000G   00              03   FB 0003C               CALLS   #3, LIB$TPARSE                                 ;
                      CC              50   E8 00043               BLBS    R0, 1$                                         ;
                          007281C4    BF   DD 00046               PUSHL   #7504324                                       ; 1755
          00000000G   00              01   FB 0004C               CALLS   #1, LIB$STOP                                   ;
                                      BD   11 00053               BRB     1$                                             ; 1749
                                      62   D5 00055 2$:           TSTL    EXT_CACHE                                      ; 1766
                                      05   12 00057               BNEQ    3$                                             ;
                E9    A2        80    BF   88 00059               BISB2   #128, MOUNT_OPTIONS+5                          ; 1768
                      01        04    A2   D1 0005E 3$:           CMPL    FID_CACHE, #1                                  ; 1770
                                      04   12 00062               BNEQ    4$                                             ;
                EA    A2              01   88 00064               BISB2   #1, MOUNT_OPTIONS+6                            ; 1772
                                08    A2   D5 00068 4$:           TSTL    QUO_CACHE                                      ; 1774
                                      04   12 0006B               BNEQ    5$                                             ;
                EA    A2              02   88 0006D               BISB2   #2, MOUNT_OPTIONS+6                            ; 1776
                                           04 00071 5$:           RET                                                    ; 1779

; Routine Size:  114 bytes,    Routine Base:  $CODE$ + 088D
; 1127          1780  1 ROUTINE DATACHECK_ACT : NOVALUE =
; 1128          1781  2 BEGIN
; 1129          1782  2
; 1130          1783  2 EXTERNAL
; 1131          1784  2         DATACHECK_STB   : VECTOR [0],    ! state table address
; 1132          1785  2         DATACHECK_KTB   : VECTOR [0];    ! keyword table address
; 1133          1786  2
; 1134          1787  2 EXTERNAL ROUTINE
; 1135          1788  2         LIB$TPARSE;
; 1136          1789  2
; 1137          1790  2 LOCAL
; 1138          1791  2     VALUE_FOUND;                         ! set when value present
; 1139          1792  2
; 1140          1793  2 ! Parse the DATACHECK options string.
; 1141          1794  2 !
; 1142          1795  2
; 1143          1796  2 VALUE_FOUND = 0;
; 1144          1797  2
; 1145          1798  2 WHILE CLI$GET_VALUE (DATA_DESC, CLI_DESC) DO
; 1146          1799  3 BEGIN
; 1147          1800  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
; 1148          1801  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
; 1149          1802  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, DATACHECK_STB, DATACHECK_KTB)
; 1150          1803  3     THEN
; 1151          1804  3         ERR_EXIT (MOUN$_BADDATCHK);
; 1152          1805  3     VALUE_FOUND = 1;
; 1153          1806  2 END;
; 1154          1807  2
; 1155          1808  2 ! If the qualifier /DATA_CHECK was specified with no value, then
; 1156          1809  2 ! WRITE data check is the default.  Set the corresponding bit.
; 1157          1810  2 !
; 1158          1811  2 IF .VALUE_FOUND EQL 0
; 1159          1812  2 THEN
; 1160          1813  2     MOUNT_OPTIONS [OPT_WRITECHECK] = 1;
; 1161          1814  2
; 1162          1815  2
; 1163          1816  1 END;                                     ! end of routine DATACHECK_ACT


                                                     .EXTRN  DATACHECK_STB, DATACHECK_KTB

                                      000C 00000 DATACHECK_ACT:
                                                     .WORD   Save R2,R3                              ; 1780
              53       0000'   CF   9E 00002         MOVAB   CLI_DESC, R3                            ;
                               52   D4 00007         CLRL    VALUE_FOUND                             ; 1796
                               53   DD 00009 1$:     PUSHL   R3                                      ; 1798
                       0000'   CF   9F 0000B         PUSHAB  DATA_DESC                               ;
      00000000G   00           02   FB 0000F         CALLS   #2, CLI$GET_VALUE                       ;
                  34           50   E9 00016         BLBC    R0, 3$                                  ;
             14   A3           63   3C 00019         MOVZWL  CLI_DESC, TPARSE_BLOCK+8                ; 1800
             18   A3    04     A3   D0 0001D         MOVL    CLI_DESC+4, TPARSE_BLOCK+12             ; 1801
                    00000000G  00   9F 00022         PUSHAB  DATACHECK_KTB                           ; 1802
                    00000000G  00   9F 00028         PUSHAB  DATACHECK_STB                           ;
                          0C   A3   9F 0002E         PUSHAB  TPARSE_BLOCK                            ;
      00000000G   00           03   FB 00031         CALLS   #3, LIB$TPARSE                          ;
                  0D           50   E8 00038         BLBS    R0, 2$                                  ;
```

```
                                          0072800C   8F  DD 0003B         PUSHL   #7503884                                    ; 1804
                            00000000G  00            01  FB 00041         CALLS   #1, LIB$STOP                                ;
                                       52            01  D0 00048 2$:     MOVL    #1, VALUE_FOUND                             ; 1805
                                                     8C  11 0004B         BRB     1$                                          ; 1798
                                                     52  D5 0004D 3$:     TSTL    VALUE_FOUND                                 ; 1811
                                                     04  12 0004F         BNEQ    4$                                          ;
                                    A4 A3            10  88 00051         BISB2   #16, MOUNT_OPTIONS+4                        ; 1813
                                                         04 00055 4$:     RET                                                 ; 1816

; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 08FF
; 1165          1817   1 ROUTINE DENSITY_ACT : NOVALUE =
; 1166          1818   1
; 1167          1819   2 BEGIN
; 1168          1820   2
; 1169          1821   2 CLI$GET_VALUE ( DENSITY_DESC, CLI_DESC );
; 1170          1822   2
; 1171          1823   3 IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
; 1172          1824   3                         .CLI_DESC [DSC$A_POINTER],
; 1173          1825   3                         DENSITY ) )
; 1174          1826   2 THEN
; 1175          1827   2     ERR_EXIT (MOUN$_BADDENS);          /
; 1176          1828   2
; 1177          1829   2 SELECTONE .DENSITY OF
; 1178          1830   2     SET
; 1179          1831   2
; 1180          1832   2     [800]        : MOUNT_OPTIONS [OPT_DENS_800] = 1;
; 1181          1833   2     [1600]       : MOUNT_OPTIONS [OPT_DENS_1600] = 1;
; 1182          1834   2     [6250]       : 1;
; 1183          1835   2     [OTHERWISE] : ERR_EXIT (MOUN$_BADDENS);
; 1184          1836   2
; 1185          1837   2     TES;
; 1186          1838   2
; 1187          1839   1 END;



                                          000C 00000 DENSITY_ACT:
                                                          .WORD   Save R2,R3                                      ; 1817
                 53 00000000G  00   9E 00002              MOVAB   LIB$STOP, R3                                    ;
                 52     0000'  CF   9E 00009              MOVAB   CLI_DESC, R2                                    ;
                               52   DD 0000E              PUSHL   R2                                              ; 1821
                        0000'  CF   9F 00010              PUSHAB  DENSITY_DESC                                    ;
       00000000G 00            02   FB 00014              CALLS   #2, CLI$GET_VALUE                               ;
                          D0   A2   9F 0001B              PUSHAB  DENSITY                                         ; 1823
                          04   A2   DD 0001E              PUSHL   CLI_DESC+4                                      ; 1824
                 7E            62   3C 00021              MOVZWL  CLI_DESC, -(SP)                                 ; 1823
       00000000G 00            03   FB 00024              CALLS   #3, LIB$CVT_DTB                                 ;
                 09            50   E8 0002B              BLBS    R0, 1$                                          ;
                    00728014   8F   DD 0002E              PUSHL   #7503892                                        ; 1827
                 63            01   FB 00034              CALLS   #1, LIB$STOP                                    ;
                 50       D0   A2   D0 00037 1$:          MOVL    DENSITY, R0                                     ; 1829
       00000320  8F            50   D1 0003B              CMPL    R0, #800                                        ; 1832
                               05   12 00042              BNEQ    2$                                              ;
             A0  A2            02   88 00044              BISB2   #2, MOUNT_OPTIONS                               ;
                                    04 00048              RET                                                     ;
       00000640  8F            50   D1 00049 2$:          CMPL    R0, #1600                                       ; 1833
                               05   12 00050              BNEQ    3$                                              ;
             A5  A2            08   88 00052              BISB2   #8, MOUNT_OPTIONS+5                             ;
                                    04 00056              RET                                                     ;
       0000186A  8F            50   D1 00057 3$:          CMPL    R0, #6250                                       ; 1834
                               09   13 0005E              BEQL    4$                                              ;
                    00728014   8F   DD 00060              PUSHL   #7503892                                        ; 1835
                 63            01   FB 00066              CALLS   #1, LIB$STOP                                    ;
                                    04 00069 4$:          RET                                                     ; 1839
```

```
; Routine Size:  106 bytes,    Routine Base:  $CODE$ + 0955

; 1188         1840  1
; 1190          1841  1 ROUTINE GET_DEVICE : NOVALUE =
; 1191          1842  1
; 1192          1843  2 BEGIN
; 1193          1844  2
; 1194          1845  2 DEVICE_COUNT = 0;
; 1195          1846  2
; 1196          1847  2 WHILE CLI$GET_VALUE ( $DESCRIPTOR('DEVICES'), CLI_DESC )
; 1197          1848  2 DO
; 1198          1849  3     BEGIN
; 1199          1850  3
; 1200          1851  3     BIND
; 1201          1852  3         DEVICE_DESC = DEVICE_STRING [.DEVICE_COUNT * 2] : $BBLOCK;
; 1202          1853  3
; 1203          1854  3     IF .DEVICE_COUNT GEQ DEVMAX
; 1204          1855  3     THEN
; 1205          1856  3         ERR_EXIT ( MOUN$_MAXDEV );
; 1206          1857  3
; 1207          1858  3     CH$FILL ( 0, DSC$C_S_BLN, DEVICE_DESC );
; 1208          1859  3     DEVICE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 1209          1860  3     DEVICE_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
; 1210          1861  3     STR$COPY_DX ( DEVICE_DESC, CLI_DESC );
; 1211          1862  3     DEVICE_COUNT = .DEVICE_COUNT + 1;
; 1212          1863  2     END;
; 1213          1864  1 END;                                          ! of routine GET_DEVICE


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                    53  45  43  49  56  45  44  00228 P.ACN:    .ASCII   \DEVICES\                                    ;
                                                    0022F           .BLKB    1
                                         00000007  00230 P.ACM:    .LONG    7                                            ;
                                         00000000' 00234           .ADDRESS P.ACN                                        ;


                                                                   .PSECT  $CODE$,NOWRT,2

                                            00FC 00000 GET_DEVICE:
                                                                   .WORD    Save R2,R3,R4,R5,R6,R7                       ; 1841
                     57        0000'   CF   9E 00002               MOVAB    DEVICE_COUNT, R7                             ;
                                       67   D4 00007               CLRL     DEVICE_COUNT                                 ; 1845
                               0170    C7   9F 00009 1$:           PUSHAB   CLI_DESC                                     ; 1847
                               0000'   CF   9F 0000D               PUSHAB   P.ACM                                        ;
             00000000G  00             02   FB 00011               CALLS    #2, CLI$GET_VALUE                            ;
                        38             50   E9 00018               BLBC     R0, 3$                                       ;
         50             67             01   78 0001B               ASHL     #1, DEVICE_COUNT, R0                         ; 1852
                        56       08 A740    DE 0001F               MOVAL    DEVICE_STRING[R0], R6                        ;
                        10             67   D1 00024               CMPL     DEVICE_COUNT, #16                            ; 1854
                                       0D   19 00027               BLSS     2$                                           ;
                         00728084      8F   DD 00029               PUSHL    #7504004                                     ; 1856
             00000000G  00             01   FB 0002F               CALLS    #1, LIB$STOP                                 ;
  08              00    6E             00   2C 00036 2$:           MOVC5    #0, (SP), #0, #8, (R6)                       ; 1858
                                       66      0003B                                                                     ;
                   02   A6     020E    8F   B0 0003C               MOVW     #526, 2(R6)                                  ; 1859
                               0170    C7   9F 00042               PUSHAB   CLI_DESC                                     ; 1861
                                       56   DD 00046               PUSHL    R6                                           ;
```

```
                          00000000G  00        02  FB 00048           CALLS   #2, STR$COPY_DX                                        ;
                                               67  D6 0004F           INCL    DEVICE_COUNT                                           ; 1862
                                               B6  11 00051           BRB     1$                                                     ; 1847
                                                   04 00053 3$:       RET                                                            ; 1864

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 09BF
; 1215        1865  1 ROUTINE GET_LABEL : NOVALUE =
; 1216        1866  1
; 1217        1867  2 BEGIN
; 1218        1868  2
; 1219        1869  2 LABEL_COUNT = 0;
; 1220        1870  2
; 1221        1871  2 WHILE CLI$GET_VALUE ( $DESCRIPTOR('VOLUMES'), CLI_DESC )
; 1222        1872  2 DO
; 1223        1873  3     BEGIN
; 1224        1874  3
; 1225        1875  3     BIND
; 1226        1876  3         LABEL_DESC = LABEL_STRING [.LABEL_COUNT * 2] : $BBLOCK;
; 1227        1877  3
; 1228        1878  3     IF .LABEL_COUNT GEQ LABMAX
; 1229        1879  3     THEN
; 1230        1880  3         ERR_EXIT ( MOUN$_MAXLAB );
; 1231        1881  3
; 1232        1882  3     CH$FILL ( 0, DSC$C_S_BLN, LABEL_DESC );
; 1233        1883  3     LABEL_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 1234        1884  3     LABEL_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
; 1235        1885  3     STR$COPY_DX ( LABEL_DESC, CLI_DESC );
; 1236        1886  3     LABEL_COUNT = .LABEL_COUNT + 1;
; 1237        1887  2     END;
; 1238        1888  1 END;                                   ! of routine GET_LABEL


                                                             .PSECT  $PLIT$,NOWRT,NOEXE,2

              53  45  4D  55  4C  4F  56  00238 P.ACP:       .ASCII  \VOLUMES\                             ;
                                          0023F              .BLKB   1
                               00000007   00240 P.ACO:       .LONG   7                                     ;
                               00000000'  00244              .ADDRESS P.ACP                                ;


                                                             .PSECT  $CODE$,NOWRT,2

                                    00FC  00000 GET_LABEL:
                                                             .WORD   Save R2,R3,R4,R5,R6,R7                ; 1865
                   57       0000'   CF  9E  00002            MOVAB   LABEL_COUNT, R7                       ;
                                    67  D4  00007            CLRL    LABEL_COUNT                           ; 1869
                            016C    C7  9F  00009 1$:        PUSHAB  CLI_DESC                              ; 1871
                            0000'   CF  9F  0000D            PUSHAB  P.ACO                                 ;
          00000000G    00           02  FB  00011            CALLS   #2, CLI$GET_VALUE                     ;
                       39           50  E9  00018            BLBC    R0, 3$                                ;
             50        67           01  78  0001B            ASHL    #1, LABEL_COUNT, R0                   ; 1876
                       56     0084  C740  DE  0001F          MOVAL   LABEL_STRING[R0], R6                  ;
                       10           67  D1  00025            CMPL    LABEL_COUNT, #16                      ; 1878
                                    0D  19  00028            BLSS    2$                                    ;
                     0072808C       8F  DD  0002A            PUSHL   #7504012                              ; 1880
          00000000G    00           01  FB  00030            CALLS   #1, LIB$STOP                          ;
    08       00        6E           00  2C  00037 2$:        MOVC5   #0, (SP), #0, #8, (R6)                ; 1882
                                    66      0003C                                                          ;
               02      A6   020E    8F  B0  0003D            MOVW    #526, 2(R6)                           ; 1883
                            016C    C7  9F  00043            PUSHAB  CLI_DESC                              ; 1885
                                    56  DD  00047            PUSHL   R6                                    ;
```

```
                    00000000G  00         02  FB 00049          CALLS   #2, STR$COPY_DX                     :
                                          67  D6 00050          INCL    LABEL_COUNT                         : 1886
                                          B5  11 00052          BRB     1$                                  : 1871
                                              04 00054 3$:      RET                                         : 1888

; Routine Size:  85 bytes,    Routine Base:  $CODE$ + 0A13
; 1240      1889  1 ROUTINE GET_LOG_NAME: NOVALUE =
; 1241      1890  1
; 1242      1891  2 BEGIN
; 1243      1892  2
; 1244      1893  2 LOCAL
; 1245      1894  2         P;                      ! string scan pointer
; 1246      1895  2
; 1247      1896  2 ! Copy the logical name descriptor
; 1248      1897  2 !
; 1249      1898  2
; 1250      1899  2 IF CLI$GET_VALUE ( $DESCRIPTOR('LOGNAMES'), CLI_DESC )
; 1251      1900  2 THEN
; 1252      1901  3     BEGIN
; 1253      1902  3     MOUNT_OPTIONS [OPT_LOG_NAME] = 1;
; 1254      1903  3     CH$FILL ( 0, DSC$C_S_BLN, LOG_NAME );
; 1255      1904  3     LOG_NAME [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 1256      1905  3     LOG_NAME [DSC$B_CLASS] = DSC$K_CLASS_D;
; 1257      1906  3     STR$COPY_DX ( LOG_NAME, CLI_DESC );
; 1258      1907  3
; 1259      1908  3     ! If logical name is greater than maximum size, return error.
; 1260      1909  3     !
; 1261      1910  3
; 1262      1911  4     IF .LOG_NAME [DSC$W_LENGTH] GTR (LOG$C_NAMLENGTH - 1)
; 1263      1912  3     THEN
; 1264      1913  3         ERR_EXIT ( MOUN$_LOGNAME );
; 1265      1914  3
; 1266      1915  3     ! Scan for a trailing of embedded colon.  If found, use string preceding
; 1267      1916  3     ! the colon.
; 1268      1917  3     !
; 1269      1918  3     P = CH$FIND_CH ( .LOG_NAME [DSC$W_LENGTH], .LOG_NAME [DSC$A_POINTER], ':' );
; 1270      1919  3
; 1271      1920  3     IF NOT CH$FAIL (.P)
; 1272      1921  3     THEN
; 1273      1922  3         LOG_NAME [DSC$W_LENGTH] = .P - .LOG_NAME [DSC$A_POINTER];
; 1274      1923  2     END;
; 1275      1924  2
; 1276      1925  1 END;                                    ! end of routine LOG_NAME_ACT


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

        53  45  4D  41  4E  47  4F  4C  00248 P.ACR:  .ASCII  \LOGNAMES\                        ;
                                00000008  00250 P.ACQ:  .LONG   8                                ;
                                00000000' 00254         .ADDRESS P.ACR                           ;


                                                                .PSECT  $CODE$,NOWRT,2

                                     007C 00000 GET_LOG_NAME:
                                                        .WORD   Save R2,R3,R4,R5,R6              ; 1889
                    56      0000'  CF  9E 00002         MOVAB   LOG_NAME, R6                     ;
                              68   A6  9F 00007         PUSHAB  CLI_DESC                         ; 1899
                            0000'  CF  9F 0000A         PUSHAB  P.ACQ                            ;
        00000000G   00             02  FB 0000E         CALLS   #2, CLI$GET_VALUE                ;
                    40             50  E9 00015         BLBC    R0, 3$                           ;
```

```
                                     0B   A6              20  88 00018             BISB2   #32, MOUNT_OPTIONS+3                         ; 1902
             08             00            6E              00  2C 0001C             MOVC5   #0, (SP), #0, #8, LOG_NAME                   ; 1903
                                                          66     00021                                                                  ;
                                     02   A6        020E  8F  B0 00022             MOVW    #526, LOG_NAME+2                             ; 1904
                                                      68  A6  9F 00028             PUSHAB  CLI_DESC                                     ; 1906
                                                          56  DD 0002B             PUSHL   R6                                           ;
                                00000000G 00              02  FB 0002D             CALLS   #2, STR$COPY_DX                              ;
                                          3F              66  B1 00034             CMPW    LOG_NAME, #63                                ; 1911
                                                          0D  1B 00037             BLEQU   1$                                           ;
                                                0072807C  8F  DD 00039             PUSHL   #7503996                                     ; 1913
                                00000000G 00              01  FB 0003F             CALLS   #1, LIB$STOP                                 ;
                        04  B6                66          3A  3A 00046 1$:         LOCC    #58, LOG_NAME, @LOG_NAME+4                   ; 1918
                                                          02  12 0004B             BNEQ    2$                                           ;
                                                          51  D4 0004D             CLRL    R1                                           ;
                                                          51  D5 0004F 2$:         TSTL    P                                            ; 1920
                                                          05  13 00051             BEQL    3$                                           ;
                            66            51          04  A6  A3 00053             SUBW3   LOG_NAME+4, P, LOG_NAME                      ; 1922
                                                              04 00058 3$:         RET                                                  ; 1925

; Routine Size:  89 bytes,    Routine Base:  $CODE$ + 0A68
; 1278       1926 1 ROUTINE INITIALIZE_ACT : NOVALUE =
; 1279       1927 1
; 1280       1928 2 BEGIN
; 1281       1929 2
; 1282       1930 2 EXTERNAL
; 1283       1931 2         INITIALIZE_STB   : VECTOR [0];   ! state table address
; 1284       1932 2         INITIALIZE_KTB   : VECTOR [0];   ! keyword table address
; 1285       1933 2
; 1286       1934 2 EXTERNAL ROUTINE
; 1287       1935 2         LIB$TPARSE;
; 1288       1936 2
; 1289       1937 2 ! Parse the INITIALIZE string and set appropriate flags.
; 1290       1938 2 !
; 1291       1939 2
; 1292       1940 2 WHILE CLI$GET_VALUE ( INITIALIZE_DESC, CLI_DESC ) DO
; 1293       1941 2 BEGIN
; 1294       1942 3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
; 1295       1943 3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
; 1296       1944 3     IF NOT LIB$TPARSE (TPARSE_BLOCK, INITIALIZE_STB, INITIALIZE_KTB)
; 1297       1945 3     THEN
; 1298       1946 3         ERR_EXIT (MOUN$_BADINIT);
; 1299       1947 2 END;
; 1300       1948 2
; 1301       1949 1 END;


                                                          .EXTRN  INITIALIZE_STB, INITIALIZE_KTB

                                       0004 00000 INITIALIZE_ACT:
                                                          .WORD   Save R2                                    ; 1926
              52       0000'   CF   9E 00002              MOVAB   CLI_DESC, R2
                               52   DD 00007 1$:          PUSHL   R2                                         ; 1940
                       0000'   CF   9F 00009              PUSHAB  INITIALIZE_DESC                            ;
    00000000G  00              02   FB 0000D              CALLS   #2, CLI$GET_VALUE                          ;
               31              50   E9 00014              BLBC    R0, 2$                                     ;
           14  A2              62   3C 00017              MOVZWL  CLI_DESC, TPARSE_BLOCK+8                   ; 1942
           18  A2          04  A2   D0 0001B              MOVL    CLI_DESC+4, TPARSE_BLOCK+12                ; 1943
                    00000000G  00   9F 00020              PUSHAB  INITIALIZE_KTB                             ; 1944
                    00000000G  00   9F 00026              PUSHAB  INITIALIZE_STB                             ;
                           0C  A2   9F 0002C              PUSHAB  TPARSE_BLOCK                               ;
    00000000G  00              03   FB 0002F              CALLS   #3, LIB$TPARSE                             ;
               CE              50   E8 00036              BLBS    R0, 1$                                     ;
                    00728224   8F   DD 00039              PUSHL   #7504420                                   ; 1946
    00000000G  00              01   FB 0003F              CALLS   #1, LIB$STOP                               ;
                               BF   11 00046              BRB     1$                                         ; 1940
                                    04 00048 2$:          RET                                                ; 1949

; Routine Size:  73 bytes,    Routine Base:  $CODE$ + 0AC1
; 1303      1950 1 ROUTINE JOURNAL_ACT : NOVALUE =
; 1304      1951 2 BEGIN
; 1305      1952 2
; 1306      1953 2 LITERAL
; 1307      1954 2     MOUNT$K_DEF_JRNL_RECORD_SIZE = 600; ! Default value for max record size
; 1308      1955 2
; 1309      1956 2 EXTERNAL
; 1310      1957 2         JOURNAL_STB     : VECTOR [0],   ! state table address
; 1311      1958 2         JOURNAL_KTB     : VECTOR [0];   ! keyword table address
; 1312      1959 2
; 1313      1960 2 EXTERNAL ROUTINE
; 1314      1961 2         LIB$TPARSE;
; 1315      1962 2
; 1316      1963 2 ! Parse the journal control options and set appropriate flags.
; 1317      1964 2 !
; 1318      1965 2 MOUNT_OPTIONS [OPT_NOJRNL] = 0;
; 1319      1966 2
; 1320      1967 2 WHILE CLI$GET_VALUE ( JOURNAL_DESC, CLI_DESC ) DO
; 1321      1968 3 BEGIN
; 1322      1969 3     TPARSE_BLOCK[TPA$L_STRINGCNT]       = .CLI_DESC[DSC$W_LENGTH];
; 1323      1970 3     TPARSE_BLOCK[TPA$L_STRINGPTR]       = .CLI_DESC[DSC$A_POINTER];
; 1324      1971 3     IF NOT LIB$TPARSE (TPARSE_BLOCK, JOURNAL_STB, JOURNAL_KTB)
; 1325      1972 3     THEN
; 1326      1973 3         ERR_EXIT (MOUN$_BADJRNL);
; 1327      1974 2 END;
; 1328      1975 2
; 1329      1976 2 ! If this is a MOUNT/JOURNAL=NEWFILE, then make sure RECORD_SIZE has a value.
; 1330      1977 2 ! Otherwise, ensure that no values were specified for journal creation
; 1331      1978 2 ! keywords.
; 1332      1979 2 !
; 1333      1980 2 IF .MOUNT_OPTIONS [OPT_NEWJRNL]
; 1334      1981 2 THEN
; 1335      1982 3     BEGIN
; 1336      1983 3     IF .JRNL_RECORD_SIZE EQL 0
; 1337      1984 3     THEN
; 1338      1985 3         JRNL_RECORD_SIZE = MOUNT$K_DEF_JRNL_RECORD_SIZE
; 1339      1986 3     END
; 1340      1987 4 ELSE IF ((.JRNL_SIZE NEQ 0) OR (.JRNL_RECORD_SIZE NEQ 0) OR (.JRNL_EXTEND NEQ 0)
; 1341      1988 3         OR (.JRNL_QUOTA NEQ 0))
; 1342      1989 2 THEN
; 1343      1990 2     ERR_EXIT (MOUN$_BADJRNL);
; 1344      1991 2
; 1345      1992 1 END;                                      ! end of routine JOURNAL_ACT


                                                         .EXTRN  JOURNAL_STB, JOURNAL_KTB

                                    000C 00000 JOURNAL_ACT:
                                                         .WORD   Save R2,R3                          ; 1950
                 53 00000000G   00   9E 00002            MOVAB   LIB$STOP, R3                        ;
                 52     0000'   CF   9E 00009            MOVAB   JRNL_RECORD_SIZE, R2                ;
           C2    A2       80    8F   8A 0000E            BICB2   #128, MOUNT_OPTIONS+6               ; 1965
                          1C    A2   9F 00013 1$:        PUSHAB  CLI_DESC                            ; 1967
                        0000'   CF   9F 00016            PUSHAB  JOURNAL_DESC                        ;
     00000000G   00             02   FB 0001A            CALLS   #2, CLI$GET_VALUE                   ;
                 2E             50   E9 00021            BLBC    R0, 2$                              ;
```

```
                30  A2         1C  A2  3C 00024         MOVZWL  CLI_DESC, TPARSE_BLOCK+8              ; 1969
                34  A2         20  A2  D0 00029         MOVL    CLI_DESC+4, TPARSE_BLOCK+12           ; 1970
                        00000000G  00  9F 0002E         PUSHAB  JOURNAL_KTB                           ; 1971
                        00000000G  00  9F 00034         PUSHAB  JOURNAL_STB                           ;
                               28  A2  9F 0003A         PUSHAB  TPARSE_BLOCK                          ;
         00000000G  00             03  FB 0003D         CALLS   #3, LIB$TPARSE                        ;
                    CC             50  E8 00044         BLBS    R0, 1$                                ;
                         00728214  8F  DD 00047         PUSHL   #7504404                              ; 1973
                    63             01  FB 0004D         CALLS   #1, LIB$STOP                          ;
                                   C1  11 00050         BRB     1$                                    ; 1967
                    0A         C3  A2  E9 00052 2$:     BLBC    MOUNT_OPTIONS+7, 3$                   ; 1980
                                   62  D5 00056         TSTL    JRNL_RECORD_SIZE                      ; 1983
                                   22  12 00058         BNEQ    5$                                    ;
                    62       0258  8F  3C 0005A         MOVZWL  #600, JRNL_RECORD_SIZE                ; 1985
                                       04 0005F         RET                                           ; 1982
                               FC  A2  D5 00060 3$:     TSTL    JRNL_SIZE                             ; 1987
                                   0E  12 00063         BNEQ    4$                                    ;
                                   62  D5 00065         TSTL    JRNL_RECORD_SIZE                      ;
                                   0A  12 00067         BNEQ    4$                                    ;
                               F8  A2  D5 00069         TSTL    JRNL_EXTEND                           ;
                                   05  12 0006C         BNEQ    4$                                    ;
                               F4  A2  D5 0006E         TSTL    JRNL_QUOTA                            ; 1988
                                   09  13 00071         BEQL    5$                                    ;
                         00728214  8F  DD 00073 4$:     PUSHL   #7504404                              ; 1990
                    63             01  FB 00079         CALLS   #1, LIB$STOP                          ;
                                       04 0007C 5$:     RET                                           ; 1992

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 0B0A
; 1347          1993    1 ROUTINE OVERRIDE_ACT : NOVALUE =
; 1348          1994    2 BEGIN
; 1349          1995    2
; 1350          1996    2 EXTERNAL
; 1351          1997    2         OVERRIDE_STB    : VECTOR [0],    ! state table address
; 1352          1998    2         OVERRIDE_KTB    : VECTOR [0];    ! keyword table address
; 1353          1999    2
; 1354          2000    2 EXTERNAL ROUTINE
; 1355          2001    2         LIB$TPARSE;
; 1356          2002    2
; 1357          2003    2 ! Parse the OVERRIDE string and set appropriate flags.
; 1358          2004    2 !
; 1359          2005    2
; 1360          2006    2 WHILE CLI$GET_VALUE ( OVERRIDE_DESC, CLI_DESC ) DO
; 1361          2007    3 BEGIN
; 1362          2008    3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
; 1363          2009    3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
; 1364          2010    3     IF NOT LIB$TPARSE (TPARSE_BLOCK, OVERRIDE_STB, OVERRIDE_KTB)
; 1365          2011    3     THEN
; 1366          2012    3         ERR_EXIT (MOUN$_BADOVR);
; 1367          2013    2 END;
; 1368          2014    2
; 1369          2015    1 END;                                        ! end of routine OVERRIDE_ACT


                                                                  .EXTRN  OVERRIDE_STB, OVERRIDE_KTB

                                                  0004 00000 OVERRIDE_ACT:
                                                                  .WORD   Save R2                              ; 1993
                         52      0000'   CF   9E 00002          MOVAB   CLI_DESC, R2
                                         52   DD 00007 1$:      PUSHL   R2                                   ; 2006
                                 0000'   CF   9F 00009          PUSHAB  OVERRIDE_DESC
             00000000G   00              02   FB 0000D          CALLS   #2, CLI$GET_VALUE
                         31              50   E9 00014          BLBC    R0, 2$
                   14    A2              62   3C 00017          MOVZWL  CLI_DESC, TPARSE_BLOCK+8             ; 2008
                   18    A2      04      A2   D0 0001B          MOVL    CLI_DESC+4, TPARSE_BLOCK+12          ; 2009
                             00000000G   00   9F 00020          PUSHAB  OVERRIDE_KTB                         ; 2010
                             00000000G   00   9F 00026          PUSHAB  OVERRIDE_STB
                                     0C  A2   9F 0002C          PUSHAB  TPARSE_BLOCK
             00000000G   00              03   FB 0002F          CALLS   #3, LIB$TPARSE
                         CE              50   E8 00036          BLBS    R0, 1$
                             0072816C    8F   DD 00039          PUSHL   #7504236                             ; 2012
             00000000G   00              01   FB 0003F          CALLS   #1, LIB$STOP
                                         BF   11 00046          BRB     1$                                   ; 2006
                                              04 00048 2$:      RET                                          ; 2015

; Routine Size:  73 bytes,    Routine Base:  $CODE$ + 0B87
; 1371          2016 1 ROUTINE OWNER_UIC_ACT : NOVALUE =
; 1372          2017 2 BEGIN
; 1373          2018 2
; 1374          2019 2 EXTERNAL
; 1375          2020 2         UIC_STB         : VECTOR [0],   ! state table address
; 1376          2021 2         UIC_KTB         : VECTOR [0];   ! keyword table address
; 1377          2022 2
; 1378          2023 2 EXTERNAL ROUTINE
; 1379          2024 2         LIB$TPARSE;
; 1380          2025 2
; 1381          2026 2 ! Parse the UIC string and store it in the owner UIC longword.
; 1382          2027 2 !
; 1383          2028 2
; 1384          2029 2 WHILE CLI$GET_VALUE ( OWNER_DESC, CLI_DESC ) DO
; 1385          2030 4 BEGIN
; 1386          2031 3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
; 1387          2032 3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
; 1388          2033 3     IF NOT LIB$TPARSE (TPARSE_BLOCK, UIC_STB, UIC_KTB)
; 1389          2034 3     THEN
; 1390          2035 3         ERR_EXIT (MOUN$_BADUIC);
; 1391          2036 2 END;
; 1392          2037 2
; 1393          2038 2 OWNER_UIC = .UIC;
; 1394          2039 2
; 1395          2040 1 END;                                    ! end of routine OWNER_UIC_ACT


                                                                          .EXTRN  UIC_STB, UIC_KTB

                                                       0004 00000 OWNER_UIC_ACT:
                                                                          .WORD   Save R2                                     ; 2016
                                       52     0000'  CF  9E 00002         MOVAB   CLI_DESC, R2
                                                     52  DD 00007 1$:     PUSHL   R2                                          ; 2029
                                              0000'  CF  9F 00009         PUSHAB  OWNER_DESC
                            00000000G  00            02  FB 0000D         CALLS   #2, CLI$GET_VALUE
                                       31            50  E9 00014         BLBC    R0, 2$
                                   14  A2            62  3C 00017         MOVZWL  CLI_DESC, TPARSE_BLOCK+8                    ; 2031
                                   18  A2        04  A2  D0 0001B         MOVL    CLI_DESC+4, TPARSE_BLOCK+12                 ; 2032
                                         00000000G   00  9F 00020         PUSHAB  UIC_KTB                                     ; 2033
                                         00000000G   00  9F 00026         PUSHAB  UIC_STB
                                                 0C  A2  9F 0002C         PUSHAB  TPARSE_BLOCK
                            00000000G  00            03  FB 0002F         CALLS   #3, LIB$TPARSE
                                       CE            50  E8 00036         BLBS    R0, 1$
                                          00728024   8F  DD 00039         PUSHL   #7503908                                    ; 2035
                            00000000G  00            01  FB 0003F         CALLS   #1, LIB$STOP
                                                     BF  11 00046         BRB     1$                                          ; 2029
                                   E8  A2        30  A2  D0 00048 2$:     MOVL    UIC, OWNER_UIC                              ; 2038
                                                         04 0004D         RET                                                 ; 2040

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 0BD0
; 1397          2041  1
; 1398          2042  1 ROUTINE PROCESSOR_ACT : NOVALUE =
; 1399          2043  2 BEGIN
; 1400          2044  2
; 1401          2045  2 EXTERNAL
; 1402          2046  2         PROCESSOR_STB   : VECTOR [0],   ! state table address
; 1403          2047  2         PROCESSOR_KTB   : VECTOR [0];   ! keyword table address
; 1404          2048  2
; 1405          2049  2 EXTERNAL ROUTINE
; 1406          2050  2         LIB$TPARSE;
; 1407          2051  2
; 1408          2052  2 ! Parse the PROCESSOR switch options (leaving values and bits set).
; 1409          2053  2 !
; 1410          2054  2
; 1411          2055  2 CLI$GET_VALUE ( PROCESSOR_DESC, CLI_DESC );
; 1412          2056  2 TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
; 1413          2057  2 TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
; 1414          2058  2
; 1415          2059  2 IF NOT LIB$TPARSE (TPARSE_BLOCK, PROCESSOR_STB, PROCESSOR_KTB)
; 1416          2060  2 THEN
; 1417          2061  2     ERR_EXIT (MOUN$_BADACP);
; 1418          2062  2
; 1419          2063  1 END;                                    ! end of routine PROCESSOR_ACT


                                                        .EXTRN  PROCESSOR_STB, PROCESSOR_KTB

                                    0004 00000 PROCESSOR_ACT:
                                                        .WORD   Save R2                                       ; 2042
                          52      0000'   CF   9E 00002         MOVAB   CLI_DESC, R2                                  ;
                                          52   DD 00007         PUSHL   R2                                            ; 2055
                                  0000'   CF   9F 00009         PUSHAB  PROCESSOR_DESC                                ;
                00000000G 00              02   FB 0000D         CALLS   #2, CLI$GET_VALUE                             ;
                       14 A2              62   3C 00014         MOVZWL  CLI_DESC, TPARSE_BLOCK+8                      ; 2056
                       18 A2      04      A2   D0 00018         MOVL    CLI_DESC+4, TPARSE_BLOCK+12                   ; 2057
                             00000000G    00   9F 0001D         PUSHAB  PROCESSOR_KTB                                 ; 2059
                             00000000G    00   9F 00023         PUSHAB  PROCESSOR_STB                                 ;
                                    0C    A2   9F 00029         PUSHAB  TPARSE_BLOCK                                  ;
                00000000G 00              03   FB 0002C         CALLS   #3, LIB$TPARSE                                ;
                          0D              50   E8 00033         BLBS    R0, 1$                                        ;
                             0072815C     8F   DD 00036         PUSHL   #7504220                                      ; 2061
                00000000G 00              01   FB 0003C         CALLS   #1, LIB$STOP                                  ;
                                               04 00043 1$:     RET                                                   ; 2063

; Routine Size:  68 bytes,    Routine Base:  $CODE$ + 0C1E
; 1421          2064  1 ROUTINE PROTECTION_ACT : NOVALUE =
; 1422          2065  1
; 1423          2066  2 BEGIN
; 1424          2067  2
; 1425          2068  2 EXTERNAL
; 1426          2069  2         PROTECTION_STB  : VECTOR [0],    ! state table address
; 1427          2070  2         PROTECTION_KTB  : VECTOR [0];    ! keyword table address
; 1428          2071  2
; 1429          2072  2 EXTERNAL ROUTINE
; 1430          2073  2         LIB$TPARSE;
; 1431          2074  2
; 1432          2075  2 ! Parse the PROTECTION qualifier string storing the binary protection.
; 1433          2076  2 ! Complement thereafter, since the parser produces the complement.
; 1434          2077  2 !
; 1435          2078  2
; 1436          2079  2 WHILE CLI$GET_VALUE ( PROTECTION_DESC, CLI_DESC ) DO
; 1437          2080  3 BEGIN
; 1438          2081  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC[DSC$W_LENGTH];
; 1439          2082  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC[DSC$A_POINTER];
; 1440          2083  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, PROTECTION_STB, PROTECTION_KTB)
; 1441          2084  3     THEN
; 1442          2085  3         ERR_EXIT (MOUN$_BADPRO);
; 1443          2086  2 END;
; 1444          2087  2
; 1445          2088  2 PROTECTION <0, 16> = NOT .PROTECTION <0, 16>;
; 1446          2089  2
; 1447          2090  1 END;                                      ! end of routine PROTECTION_ACT


                                                          .EXTRN  PROTECTION_STB, PROTECTION_KTB

                                       0004 00000 PROTECTION_ACT:
                                                          .WORD   Save R2                                      ; 2064
                     52      0000'   CF  9E 00002         MOVAB   CLI_DESC, R2                                 ;
                                     52  DD 00007 1$:     PUSHL   R2                                           ; 2079
                             0000'   CF  9F 00009         PUSHAB  PROTECTION_DESC                              ;
           00000000G 00              02  FB 0000D         CALLS   #2, CLI$GET_VALUE                            ;
                     31              50  E9 00014         BLBC    R0, 2$                                       ;
                14   A2              62  3C 00017         MOVZWL  CLI_DESC, TPARSE_BLOCK+8                     ; 2081
                18   A2        04    A2  D0 0001B         MOVL    CLI_DESC+4, TPARSE_BLOCK+12                  ; 2082
                         00000000G   00  9F 00020         PUSHAB  PROTECTION_KTB                               ; 2083
                         00000000G   00  9F 00026         PUSHAB  PROTECTION_STB                               ;
                               0C    A2  9F 0002C         PUSHAB  TPARSE_BLOCK                                 ;
           00000000G 00              03  FB 0002F         CALLS   #3, LIB$TPARSE                               ;
                     CE              50  E8 00036         BLBS    R0, 1$                                       ;
                          0072801C   8F  DD 00039         PUSHL   #7503900                                     ; 2085
           00000000G 00              01  FB 0003F         CALLS   #1, LIB$STOP                                 ;
                                     BF  11 00046         BRB     1$                                           ; 2079
                EC   A2        EC    A2  B2 00048 2$:     MCOMW   PROTECTION, PROTECTION                       ; 2088
                                         04 0004D         RET                                                  ; 2090

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 0C62
; 1449          2091  1
; 1450          2092  1
; 1451          2093  1 !+
; 1452          2094  1 !
; 1453          2095  1 !  TPARSE action routines for the following TPARSE tables.
; 1454          2096  1 !
; 1455          2097  1 !-
; 1456          2098  1
; 1457          2099  1 !
; 1458          2100  1 !  Clear the "NEW JOURNAL FILE" option bit.  (We just saw NONEWFILE.)
; 1459          2101  1 !
; 1460          2102  1 ROUTINE CLEAR_NEWJRNL =
; 1461          2103  2 BEGIN
; 1462          2104  2
; 1463          2105  2 MOUNT_OPTIONS [OPT_NEWJRNL] = 0;
; 1464          2106  2 RETURN 1;
; 1465          2107  2
; 1466          2108  1 END;



                                                0000 00000 CLEAR_NEWJRNL:
                                                                 .WORD   Save nothing                                  ; 2102
                         0000'  CF              01  8A 00002     BICB2   #1, MOUNT_OPTIONS+7                           ; 2105
                                50              01  D0 00007     MOVL    #1, R0                                        ; 2106
                                                    04 0000A     RET                                                   ; 2108

; Routine Size:  11 bytes,    Routine Base:  $CODE$ + 0CB0


; 1467          2109  1
; 1469          2110  1 !
; 1470          2111  1 ! Store ACP string (either device name or file name).
; 1471          2112  1 !
; 1472          2113  1 ROUTINE GET_ACP_NAME =
; 1473          2114  1
; 1474          2115  2 BEGIN
; 1475          2116  2
; 1476          2117  2 LOCAL
; 1477          2118  2     ACP_DESC :  BBLOCK [DSC$C_S_BLN];
; 1478          2119  2
; 1479          2120  2 TPARSE_ARGS (CONTEXT);
; 1480          2121  2
; 1481          2122  2 IF .CONTEXT[TPA$L_TOKENCNT] GTR 20
; 1482          2123  2 THEN ERR_EXIT (MOUN$_ACPNAME);
; 1483          2124  2
; 1484          2125  2 ! Initialize local descriptor and load values
; 1485          2126  2 !
; 1486          2127  2 CH$FILL ( 0, DSC$C_S_BLN, ACP_DESC);
; 1487          2128  2 ACP_DESC [DSC$B_DTYPE] = DSC$K_CLASS_D;
; 1488          2129  2 ACP_DESC [DSC$W_LENGTH] = .CONTEXT [TPA$L_TOKENCNT];
; 1489          2130  2 ACP_DESC [DSC$A_POINTER] = .CONTEXT [TPA$L_TOKENPTR];
; 1490          2131  2
; 1491          2132  2 ! Now, move values to the text descriptor.  We need to use a temporary
; 1492          2133  2 ! ACP descriptor, because the CLI_DESC contains the keyword 'SAME:'
; 1493          2134  2 ! when that option is used.
; 1494          2135  2 !
; 1495          2136  2 CH$FILL ( 0, DSC$C_S_BLN, ACP_STRING );
; 1496          2137  2 ACP_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 1497          2138  2 ACP_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
; 1498          2139  2 STR$COPY_DX ( ACP_STRING, ACP_DESC );
; 1499          2140  2 RETURN 1;
; 1500          2141  2
; 1501          2142  1 END;                                  ! end of routine GET_ACP_NAME




                                                003C 00000 GET_ACP_NAME:
                                                                  .WORD   Save R2,R3,R4,R5                              ; 2113
                           5E              08   C2 00002          SUBL2   #8, SP                                        ;
                           14        10    AC   D1 00005          CMPL    16(CONTEXT), #20                              ; 2122
                                           0D   15 00009          BLEQ    1$                                            ;
                              00728144     8F   DD 0000B          PUSHL   #7504196                                      ; 2123
                 00000000G 00              01   FB 00011          CALLS   #1, LIB$STOP                                  ;
      08           00      6E              00   2C 00018 1$:      MOVC5   #0, (SP), #0, #8, ACP_DESC                    ; 2127
                                           6E      0001D                                                                ;
                       02  AE              02   90 0001E          MOVB    #2, ACP_DESC+2                                ; 2128
                           6E        10    AC   B0 00022          MOVW    16(CONTEXT), ACP_DESC                         ; 2129
                       04  AE        14    AC   D0 00026          MOVL    20(CONTEXT), ACP_DESC+4                       ; 2130
      08           00      6E              00   2C 0002B          MOVC5   #0, (SP), #0, #8, ACP_STRING                  ; 2136
                                  0000'    CF      00030                                                                ;
                    0000'  CF     020E     8F   B0 00033          MOVW    #526, ACP_STRING+2                            ; 2137
                                           5E   DD 0003A          PUSHL   SP                                            ; 2139
                                  0000'    CF   9F 0003C          PUSHAB  ACP_STRING                                    ;
                 00000000G 00              02   FB 00040          CALLS   #2, STR$COPY_DX                               ;
                           50              01   D0 00047          MOVL    #1, R0                                        ; 2140
```

```
                                           04 0004A         RET                                              ; 2142

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 0CBB
; 1503          2143  1  !
; 1504          2144  1  ! Store ACP string as specified by the :SAME option.
; 1505          2145  1  ! Append a ":" to the device name.
; 1506          2146  1  !
; 1507          2147  1  ROUTINE GET_SAME_ACP =
; 1508          2148  2  BEGIN
; 1509          2149  2
; 1510          2150  2  LOCAL
; 1511          2151  2      ACP_DESC :  BBLOCK [DSC$C_S_BLN],
; 1512          2152  2      SAME_ACP : VECTOR [21,BYTE];
; 1513          2153  2
; 1514          2154  2  TPARSE_ARGS (CONTEXT);
; 1515          2155  2
; 1516          2156  2  IF .CONTEXT[TPA$L_TOKENCNT] GTR 20
; 1517          2157  2  THEN ERR_EXIT (MOUN$_ACPNAME);
; 1518          2158  2
; 1519          2159  2  ! Add the colon (:) to the device name.
; 1520          2160  2  !
; 1521          2161  2  CH$MOVE (.CONTEXT[TPA$L_TOKENCNT], .CONTEXT[TPA$L_TOKENPTR], SAME_ACP);
; 1522          2162  2  SAME_ACP [.CONTEXT [TPA$L_TOKENCNT]] = %ASCII ':';
; 1523          2163  2
; 1524          2164  2  ! Initialize local descriptor and load values.  The size of the device
; 1525          2165  2  ! name has increased by 1, because of the colon that was added.
; 1526          2166  2  !
; 1527          2167  2  CH$FILL ( 0, DSC$C_S_BLN, ACP_DESC);
; 1528          2168  2  ACP_DESC [DSC$B_DTYPE] = DSC$K_CLASS_D;
; 1529          2169  2  ACP_DESC [DSC$W_LENGTH] = .CONTEXT [TPA$L_TOKENCNT] + 1;
; 1530          2170  2  ACP_DESC [DSC$A_POINTER] = SAME_ACP;
; 1531          2171  2
; 1532          2172  2  ! Now, move values to the text descriptor.
; 1533          2173  2  !
; 1534          2174  2  CH$FILL ( 0, DSC$C_S_BLN, ACP_STRING );
; 1535          2175  2  ACP_STRING [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 1536          2176  2  ACP_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
; 1537          2177  2  STR$COPY_DX ( ACP_STRING, ACP_DESC );
; 1538          2178  2  RETURN 1;
; 1539          2179  2
; 1540          2180  1  END;                                        ! end of routine GET_SAME_ACP



                                                   003C 00000 GET_SAME_ACP:
                                                                     .WORD   Save R2,R3,R4,R5                               ; 2147
                          5E                  20   C2 00002          SUBL2   #32, SP                                        ;
                          14          10      AC   D1 00005          CMPL    16(CONTEXT), #20                               ; 2156
                                              0D   15 00009          BLEQ    1$                                             ;
                               00728144       8F   DD 0000B          PUSHL   #7504196                                       ; 2157
                000000000G  00                01   FB 00011          CALLS   #1, LIB$STOP                                   ;
              6E      14    BC        10      AC   28 00018 1$:      MOVC3   16(CONTEXT), @20(CONTEXT), SAME_ACP            ; 2161
                            50                6E   9E 0001E          MOVAB   SAME_ACP, R0                                   ; 2162
                      10 BC40                 3A   90 00021          MOVB    #58, @16(CONTEXT)[R0]                          ;
   08         00            6E                00   2C 00026          MOVC5   #0, (SP), #0, #8, ACP_DESC                     ; 2167
                                      18      AE      0002B                                                                 ;
                      1A    AE                02   90 0002D          MOVB    #2, ACP_DESC+2                                 ; 2168
           18  AE     10    AC                01   A1 00031          ADDW3   #1, 16(CONTEXT), ACP_DESC                      ; 2169
```

```
                                   1C   AE              6E  9E 00037          MOVAB    SAME_ACP, ACP_DESC+4                             ; 2170
        08              00              6E              0C  2C 0003B          MOVC5    #0, (SP), #0, #8, ACP_STRING                     ; 2174
                                                 0000'  CF     00040                                                                    ;
                               0000'    CF       020E   8F  B0 00043          MOVW     #526, ACP_STRING+2                               ; 2175
                                                   18   AE  9F 0004A          PUSHAB   ACP_DESC                                         ; 2177
                                                 0000'  CF  9F 0004D          PUSHAB   ACP_STRING                                       ;
                           00000000G    00              02  FB 00051          CALLS    #2, STR$COPY_DX                                  ;
                                        50              01  D0 00058          MOVL     #1, R0                                           ; 2178
                                                            04 0005B          RET                                                       ; 2180

; Routine Size:  92 bytes,     Routine Base:  $CODE$ + 0D06
```

```
:     1 !+
:     1 !
:     1 ! TPARSE state tables to parse the various qualifier value strings.
:     1 !
:     1 !-
:     1
:     1 !
:     1 ! Parse /CACHE options (EXTENT=n, LIMIT=n, FILE_ID=n, QUOTA=n, NOEXTENT,
:     1 ! NOFILE_ID, NOQUOTA, and WRITETHROUGH).
:     1 !
:     1 $INIT_STATE (CACHE_STB, CACHE_KTB);
:     1
: P   1 $STATE  (NEXT_CACHE,
: P   1         ('EXTENT',        CACHE_EXT,,     1^(OPT_CACHE-32), MOUNT_OPTIONS+4),
: P   1         ('FILE_ID',       CACHE_FID,,     1^(OPT_CACHE-32), MOUNT_OPTIONS+4),
: P   1         ('LIMIT',         LIMIT_EXT),
: P   1         ('NOEXTENT',,,                    1^(OPT_NOEXT_C-32), MOUNT_OPTIONS+4),
: P   1         ('NOFILE_ID',,,                   1^(OPT_NOFID_C-32), MOUNT_OPTIONS+4),
: P   1         ('NOQUOTA',,,                     1^(OPT_NOQUO_C-32), MOUNT_OPTIONS+4),
: P   1         ('NOWRITETHROUGH'),
: P   1         ('QUOTA',         CACHE_QUO,,     1^(OPT_CACHE-32), MOUNT_OPTIONS+4),
: P   1         ('WRITETHROUGH',,,                1^(OPT_WTHRU-32), MOUNT_OPTIONS+4)
:     1         );
:     1
: P   1 $STATE  (END_CACHE,
: P   1         (',', NEXT_CACHE),
: P   1         (TPA$_EOS, TPA$_EXIT)
:     1         );
:     1
: P   1 $STATE  (CACHE_EXT,
: P   1         (':'),
: P   1         ('=')
:     1         );
:     1
: P   1 $STATE  (,
: P   1         (TPA$_DECIMAL,  END_CACHE,,,    EXT_CACHE)
:     1         );
:     1
:     1
: P   1 $STATE  (CACHE_FID,
: P   1         (':'),
: P   1         ('=')
:     1         );
:     1
: P   1 $STATE  (,
: P   1         (TPA$_DECIMAL,  END_CACHE,,,    FID_CACHE)
:     1         );
:     1
:     1
: P   1 $STATE  (CACHE_QUO,
: P   1         (':'),
: P   1         ('=')
:     1         );
:     1
: P   1 $STATE  (,
: P   1         (TPA$_DECIMAL,  END_CACHE,,,    QUO_CACHE)
:     1         );
```

```
  1
P 1 $STATE  (LIMIT_EXT,
P 1         (':'),
P 1         ('=')
  1         );
  1
P 1 $STATE  (,
P 1         (TPA$_DECIMAL,  END_CACHE,,,    EXT_LIMIT)
  1         );
  1
  1 !
  1 ! Parse /DATA_CHECK options, of the form [READ][,WRITE]. Default is write.
  1 !
  1 $INIT_STATE (DATACHECK_STB, DATACHECK_KTB);
  1
P 1 $STATE  (,
P 1         (TPA$_EOS, TPA$_EXIT,, 1^(OPT_WRITECHECK-32), MOUNT_OPTIONS+4),
P 1         (TPA$_LAMBDA)
  1         );
  1
P 1 $STATE  (CHECKOPT,
P 1         ('READ',,, 1^(OPT_READCHECK-32), MOUNT_OPTIONS+4),
P 1         ('WRITE',,, 1^(OPT_WRITECHECK-32), MOUNT_OPTIONS+4)
  1         );
  1
P 1 $STATE  (,
P 1         (',', CHECKOPT),
P 1         (TPA$_EOS, TPA$_EXIT)
  1         );
  1
  1 !
  1 ! Parse INITIALIZE options (ALL, CONTINUATION)
  1 !
  1
  1 $INIT_STATE (INITIALIZE_STB, INITIALIZE_KTB);
  1
P 1 $STATE  (NEXTINI,
P 1         ('ALL',,,1^(OPT_INIT_ALL-32), MOUNT_OPTIONS+4),
P 1         ('CONTINUATION',,,1^(OPT_INIT_CONT-32), MOUNT_OPTIONS+4)
  1         );
  1
P 1 $STATE  (,
P 1         (',',NEXTINI),
P 1         (TPA$_EOS, TPA$_EXIT)
  1         );
  1 !
  1 ! Parse JOURNAL options ([NO]NEWFILE, SIZE=n, EXTENSION=n, QUOTA=n, RECORD_SIZE=n)
  1 !
  1 $INIT_STATE (JOURNAL_STB, JOURNAL_KTB);
  1
P 1 $STATE  (NEXT_JOURNAL,
P 1         ('NEWFILE',,,               1^(OPT_NEWJRNL-32), MOUNT_OPTIONS+4),
P 1         ('NONEWFILE',,         CLEAR_NEWJRNL),
P 1         ('SIZE',        JOURNAL_SIZE),
P 1         ('RECORD_SIZE', JOURNAL_RECORD_SIZE),
P 1         ('EXTENSION',   JOURNAL_EXTEND),
P 1         ('QUOTA',       JOURNAL_QUOTA),
; 1656        P 2295   1          (TPA$_EOS,      TPA$_EXIT)
; 1657          2296   1          );
; 1658          2297   1
; 1659        P 2298   1 $STATE   (END_JOURNAL,
; 1660        P 2299   1          (',',           NEXT_JOURNAL),
; 1661        P 2300   1          (TPA$_EOS,      TPA$_EXIT)
; 1662          2301   1          );
; 1663          2302   1
; 1664        P 2303   1 $STATE   (JOURNAL_SIZE,
; 1665        P 2304   1          (':'),
; 1666        P 2305   1          ('=')
; 1667          2306   1          );
; 1668          2307   1
; 1669        P 2308   1 $STATE   (,
; 1670        P 2309   1          (TPA$_DECIMAL,  END_JOURNAL,,,  JRNL_SIZE)
; 1671          2310   1          );
; 1672          2311   1
; 1673        P 2312   1 $STATE   (JOURNAL_RECORD_SIZE,
; 1674        P 2313   1          (':'),
; 1675        P 2314   1          ('=')
; 1676          2315   1          );
; 1677          2316   1
; 1678        P 2317   1 $STATE   (,
; 1679        P 2318   1          (TPA$_DECIMAL,  END_JOURNAL,,,  JRNL_RECORD_SIZE)
; 1680          2319   1          );
; 1681          2320   1
; 1682        P 2321   1 $STATE   (JOURNAL_EXTEND,
; 1683        P 2322   1          (':'),
; 1684        P 2323   1          ('=')
; 1685          2324   1          );
; 1686          2325   1
; 1687        P 2326   1 $STATE   (,
; 1688        P 2327   1          (TPA$_DECIMAL,  END_JOURNAL,,,  JRNL_EXTEND)
; 1689          2328   1          );
; 1690          2329   1
; 1691        P 2330   1 $STATE   (JOURNAL_QUOTA,
; 1692        P 2331   1          (':'),
; 1693        P 2332   1          ('=')
; 1694          2333   1          );
; 1695          2334   1
; 1696        P 2335   1 $STATE   (,
; 1697        P 2336   1          (TPA$_DECIMAL,  END_JOURNAL,,,  JRNL_QUOTA)
; 1698          2337   1          );
; 1699          2338   1
; 1700          2339   1 !
; 1701          2340   1 ! Parse /OVERRIDE options (ACCESSIBILITY, EXPIRATION, SETIDENTIFICATION,
; 1702          2341   1 !                         IDENTIFICATION, OWNER_IDENTIFIER).
; 1703          2342   1 !
; 1704          2343   1 $INIT_STATE (OVERRIDE_STB, OVERRIDE_KTB);
; 1705          2344   1
; 1706        P 2345   1 $STATE   (NEXTOVR,
; 1707        P 2346   1          ('ACCESSIBILITY',,,1^(OPT_OVR_ACC-32), MOUNT_OPTIONS+4),
; 1708        P 2347   1          ('EXPIRATION',,,1^OPT_OVR_EXP, MOUNT_OPTIONS),
; 1709        P 2348   1          ('SETIDENTIFICATION',,,1^OPT_OVR_SETID, MOUNT_OPTIONS),
; 1710        P 2349   1          ('LOCK',,,1^(OPT_OVR_LOCK-32), MOUNT_OPTIONS+4),
; 1711        P 2350   1          ('IDENTIFICATION',,,1^OPT_OVR_ID, MOUNT_OPTIONS),
; 1712        P 2351   1          ('OWNER_IDENTIFIER',,,1^(OPT_OVR_VOLO-32), MOUNT_OPTIONS+4)
; 1713          2352    1               );
; 1714          2353    1
; 1715        P 2354    1       $STATE  (,
; 1716        P 2355    1               (',',NEXTOVR),
; 1717        P 2356    1               (TPA$_EOS, TPA$_EXIT)
; 1718          2357    1               );
; 1719          2358    1
; 1720          2359    1
; 1721          2360    1       !
; 1722          2361    1       ! Parse /OWNER_UIC string and store binary value.
; 1723          2362    1       !
; 1724          2363    1       $INIT_STATE (UIC_STB, UIC_KTB);
; 1725          2364    1
; 1726        P 2365    1       $STATE  (,
; 1727        P 2366    1               (TPA$_IDENT,,,,UIC)
; 1728          2367    1               );
; 1729          2368    1
; 1730        P 2369    1       $STATE  (,
; 1731        P 2370    1               (TPA$_EOS, TPA$_EXIT)
; 1732          2371    1               );
; 1733          2372    1
; 1734          2373    1       !
; 1735          2374    1       ! Parse PROCESSOR options, set bits and store name.
; 1736          2375    1       !
; 1737          2376    1       $INIT_STATE (PROCESSOR_STB, PROCESSOR_KTB);
; 1738          2377    1
; 1739        P 2378    1       $STATE  (,
; 1740        P 2379    1               ('UNIQUE',, GET_ACP_NAME, 1^OPT_UNIQUEACP, MOUNT_OPTIONS),
; 1741        P 2380    1               ('SAME', SAMEPROC,, 1^OPT_SAMEACP, MOUNT_OPTIONS),
; 1742        P 2381    1               ((FILENAME),, GET_ACP_NAME, 1^OPT_FILEACP, MOUNT_OPTIONS)
; 1743          2382    1               );
; 1744          2383    1
; 1745        P 2384    1       $STATE  (ENDPROC,
; 1746        P 2385    1               (TPA$_EOS, TPA$_EXIT)
; 1747          2386    1               );
; 1748          2387    1
; 1749        P 2388    1       $STATE  (SAMEPROC,
; 1750        P 2389    1               (':'),
; 1751        P 2390    1               ('=')
; 1752          2391    1               );
; 1753          2392    1
; 1754        P 2393    1       $STATE  (,
; 1755        P 2394    1               ((DEVICENAME),, GET_ACP_NAME),
; 1756        P 2395    1               (TPA$_SYMBOL,, GET_SAME_ACP)
; 1757          2396    1               );
; 1758          2397    1
; 1759        P 2398    1       $STATE  (,
; 1760        P 2399    1               (TPA$_LAMBDA, TPA$_EXIT)
; 1761          2400    1               );
; 1762          2401    1
; 1763        P 2402    1       $STATE  (FILENAME,
; 1764        P 2403    1               (TPA$_SYMBOL, FILENAME),
; 1765        P 2404    1               ('.', FILENAME),
; 1766        P 2405    1               (':', FILENAME),
; 1767        P 2406    1               (TPA$_LAMBDA, TPA$_EXIT)
; 1768          2407    1               );
; 1769          2408    1
```

```
$STATE  (DEVICENAME,
        (TPA$_SYMBOL)
        );

$STATE  (,
        (':')
        );

$STATE  (,
        (TPA$_EOS, TPA$_EXIT)
        );

!
! Parse /PROTECTION string "(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:RWED)"
!
$INIT_STATE (PROTECTION_STB, PROTECTION_KTB);

$STATE  (NEXTPRO,
        ('SYSTEM', SYPR,, %X'000F0000', PROTECTION),
        ('OWNER',  OWPR,, %X'00F00000', PROTECTION),
        ('GROUP',  GRPR,, %X'0F000000', PROTECTION),
        ('WORLD',  WOPR,, %X'F0000000', PROTECTION)
        );

$STATE  (SYPR,
        (':'),
        ('='),
        (TPA$_LAMBDA, ENDPRO)
        );

$STATE  (SYPRO,
        ('R', SYPRO,, %X'0001', PROTECTION),
        ('W', SYPRO,, %X'0002', PROTECTION),
        ('E', SYPRO,, %X'0004', PROTECTION),
        ('P', SYPRO,, %X'0004', PROTECTION),
        ('D', SYPRO,, %X'0008', PROTECTION),
        ('L', SYPRO,, %X'0008', PROTECTION),
        (TPA$_LAMBDA, ENDPRO)
        );

$STATE  (OWPR,
        (':'),
        ('='),
        (TPA$_LAMBDA, ENDPRO)
        );

$STATE  (OWPRO,
        ('R', OWPRO,, %X'0010', PROTECTION),
        ('W', OWPRO,, %X'0020', PROTECTION),
        ('E', OWPRO,, %X'0040', PROTECTION),
        ('P', OWPRO,, %X'0040', PROTECTION),
        ('D', OWPRO,, %X'0080', PROTECTION),
        ('L', OWPRO,, %X'0080', PROTECTION),
        (TPA$_LAMBDA, ENDPRO)
        );

$STATE  (GRPR,
; 1827          P 2466  1               (':'),
; 1828          P 2467  1               ('='),
; 1829          P 2468  1               (TPA$_LAMBDA, ENDPRO)
; 1830            2469  1               );
; 1831            2470  1
; 1832          P 2471  1 $STATE        (GRPRO,
; 1833          P 2472  1               ('R', GRPRO,, %X'0100', PROTECTION),
; 1834          P 2473  1               ('W', GRPRO,, %X'0200', PROTECTION),
; 1835          P 2474  1               ('E', GRPRO,, %X'0400', PROTECTION),
; 1836          P 2475  1               ('P', GRPRO,, %X'0400', PROTECTION),
; 1837          P 2476  1               ('D', GRPRO,, %X'0800', PROTECTION),
; 1838          P 2477  1               ('L', GRPRO,, %X'0800', PROTECTION),
; 1839          P 2478  1               (TPA$_LAMBDA, ENDPRO)
; 1840            2479  1               );
; 1841            2480  1
; 1842          P 2481  1 $STATE        (WOPR,
; 1843          P 2482  1               (':'),
; 1844          P 2483  1               ('='),
; 1845          P 2484  1               (TPA$_LAMBDA, ENDPRO)
; 1846            2485  1               );
; 1847            2486  1
; 1848          P 2487  1 $STATE        (WOPRO,
; 1849          P 2488  1               ('R', WOPRO,, %X'1000', PROTECTION),
; 1850          P 2489  1               ('W', WOPRO,, %X'2000', PROTECTION),
; 1851          P 2490  1               ('E', WOPRO,, %X'4000', PROTECTION),
; 1852          P 2491  1               ('P', WOPRO,, %X'4000', PROTECTION),
; 1853          P 2492  1               ('D', WOPRO,, %X'8000', PROTECTION),
; 1854          P 2493  1               ('L', WOPRO,, %X'8000', PROTECTION),
; 1855          P 2494  1               (TPA$_LAMBDA, ENDPRO)
; 1856            2495  1               );
; 1857            2496  1
; 1858          P 2497  1 $STATE        (ENDPRO,
; 1859          P 2498  1               (',', NEXTPRO),
; 1860          P 2499  1               (TPA$_EOS, TPA$_EXIT)
; 1861            2500  1               );
; 1862            2501  1
; 1863            2502  1 END
; 1864            2503  0 ELUDOM


                                                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                                00000 ;TPA$KEYST0
                                                      U.2:      .BLKB   0
                        54  4E  45  54  58  45  00000 ;TPA$KEYST
                                                      U.4:      .ASCII  \EXTENT\                                        ;
                                            FF  00006           .BYTE   -1                                              ;
                                                00007 ;TPA$KEYST0
                                                      U.10:     .BLKB   0
                    44  49  5F  45  4C  49  46  00007 ;TPA$KEYST
                                                      U.12:     .ASCII  \FILE_ID\                                       ;
                                            FF  0000E           .BYTE   -1                                              ;
                                                0000F ;TPA$KEYST0
                                                      U.18:     .BLKB   0
                            54  49  4D  49  4C  0000F ;TPA$KEYST
                                                      U.20:     .ASCII  \LIMIT\                                         ;
```

```
                                                    FF  00014               .BYTE   -1                                  ;
                                                        00015 ;TPA$KEYST0
                                                              U.24:     .BLKB   0
                        54  4E  45  54  58  45  4F  4E  00015 ;TPA$KEYST
                                                              U.26:     .ASCII  \NOEXTENT\                          ;
                                                    FF  0001D               .BYTE   -1                                  ;
                                                        0001E ;TPA$KEYST0
                                                              U.30:     .BLKB   0
                    44  49  5F  45  4C  49  46  4F  4E  0001E ;TPA$KEYST
                                                              U.32:     .ASCII  \NOFILE_ID\                         ;
                                                    FF  00027               .BYTE   -1                                  ;
                                                        00028 ;TPA$KEYST0
                                                              U.36:     .BLKB   0
                            41  54  4F  55  51  4F  4E  00028 ;TPA$KEYST
                                                              U.38:     .ASCII  \NOQUOTA\                           ;
                                                    FF  0002F               .BYTE   -1                                  ;
                                                        00030 ;TPA$KEYST0
                                                              U.42:     .BLKB   0
48  47  55  4F  52  48  54  45  54  49  52  57  4F  4E  00030 ;TPA$KEYST
                                                              U.44:     .ASCII  \NOWRITETHROUGH\                    ;
                                                    FF  0003E               .BYTE   -1                                  ;
                                                        0003F ;TPA$KEYST0
                                                              U.46:     .BLKB   0
                                    41  54  4F  55  51  0003F ;TPA$KEYST
                                                              U.48:     .ASCII  \QUOTA\                             ;
                                                    FF  00044               .BYTE   -1                                  ;
                                                        00045 ;TPA$KEYST0
                                                              U.54:     .BLKB   0
        48  47  55  4F  52  48  54  45  54  49  52  57  00045 ;TPA$KEYST
                                                              U.56:     .ASCII  \WRITETHROUGH\                      ;
                                                    FF  00051               .BYTE   -1                                  ;
                                                    FF  00052 ;TPA$KEYFILL
                                                              U.60:     .BYTE   -1                                  ;
                                                        00053 ;TPA$KEYST0
                                                              U.91:     .BLKB   0
                                        44  41  45  52  00053 ;TPA$KEYST
                                                              U.93:     .ASCII  \READ\                              ;
                                                    FF  00057               .BYTE   -1                                  ;
                                                        00058 ;TPA$KEYST0
                                                              U.97:     .BLKB   0
                                    45  54  49  52  57  00058 ;TPA$KEYST
                                                              U.99:     .ASCII  \WRITE\                             ;
                                                    FF  0005D               .BYTE   -1                                  ;
                                                    FF  0005E ;TPA$KEYFILL
                                                              U.103:    .BYTE   -1                                  ;
                                                        0005F ;TPA$KEYST0
                                                              U.109:    .BLKB   0
                                            4C  4C  41  0005F ;TPA$KEYST
                                                              U.111:    .ASCII  \ALL\                               ;
                                                    FF  00062               .BYTE   -1                                  ;
                                                        00063 ;TPA$KEYST0
                                                              U.115:    .BLKB   0
        4E  4F  49  54  41  55  4E  49  54  4E  4F  43  00063 ;TPA$KEYST
                                                              U.117:    .ASCII  \CONTINUATION\                      ;
                                                    FF  0006F               .BYTE   -1                                  ;
                                                    FF  00070 ;TPA$KEYFILL
                                                              U.121:    .BYTE   -1                                  ;
```

```
                                             00071 ;TPA$KEYST0
                                                    U.127:   .BLKB   0
                        45 4C 49 46 57 45 4E 00071 ;TPA$KEYST
                                                    U.129:   .ASCII  \NEWFILE\                                ;
                                          FF 00078           .BYTE   -1                                       ;
                                             00079 ;TPA$KEYST0
                                                    U.133:   .BLKB   0
                  45 4C 49 46 57 45 4E 4F 4E 00079 ;TPA$KEYST
                                                    U.135:   .ASCII  \NONEWFILE\                              ;
                                          FF 00082           .BYTE   -1                                       ;
                                             00083 ;TPA$KEYST0
                                                    U.138:   .BLKB   0
                                 45 5A 49 53 00083 ;TPA$KEYST
                                                    U.140:   .ASCII  \SIZE\                                   ;
                                          FF 00087           .BYTE   -1                                       ;
                                             00088 ;TPA$KEYST0
                                                    U.144:   .BLKB   0
            45 5A 49 53 5F 44 52 4F 43 45 52 00088 ;TPA$KEYST
                                                    U.146:   .ASCII  \RECORD_SIZE\                            ;
                                          FF 00093           .BYTE   -1                                       ;
                                             00094 ;TPA$KEYST0
                                                    U.150:   .BLKB   0
                  4E 4F 49 53 4E 45 54 58 45 00094 ;TPA$KEYST
                                                    U.152:   .ASCII  \EXTENSION\                              ;
                                          FF 0009D           .BYTE   -1                                       ;
                                             0009E ;TPA$KEYST0
                                                    U.156:   .BLKB   0
                              41 54 4F 55 51 0009E ;TPA$KEYST
                                                    U.158:   .ASCII  \QUOTA\                                  ;
                                          FF 000A3           .BYTE   -1                                       ;
                                          FF 000A4 ;TPA$KEYFILL
                                                    U.164:   .BYTE   -1                                       ;
                                             000A5 ;TPA$KEYST0
                                                    U.190:   .BLKB   0
      59 54 49 4C 49 42 49 53 53 45 43 43 41 000A5 ;TPA$KEYST
                                                    U.192:   .ASCII  \ACCESSIBILITY\                          ;
                                          FF 000B2           .BYTE   -1                                       ;
                                             000B3 ;TPA$KEYST0
                                                    U.196:   .BLKB   0
               4E 4F 49 54 41 52 49 50 58 45 000B3 ;TPA$KEYST
                                                    U.198:   .ASCII  \EXPIRATION\                             ;
                                          FF 000BD           .BYTE   -1                                       ;
                                             000BE ;TPA$KEYST0
                                                    U.202:   .BLKB   0
49 54 41 43 49 46 49 54 4E 45 44 49 54 45 53 000BE ;TPA$KEYST
                                                    U.204:   .ASCII  \SETIDENTIFICATION\                      ;
                                       4E 4F 000CD                                                            ;
                                          FF 000CF           .BYTE   -1                                       ;
                                             000D0 ;TPA$KEYST0
                                                    U.208:   .BLKB   0
                                 4B 43 4F 4C 000D0 ;TPA$KEYST
                                                    U.210:   .ASCII  \LOCK\                                   ;
                                          FF 000D4           .BYTE   -1                                       ;
                                             000D5 ;TPA$KEYST0
                                                    U.214:   .BLKB   0
   4E 4F 49 54 41 43 49 46 49 54 4E 45 44 49 000D5 ;TPA$KEYST
                                                    U.216:   .ASCII  \IDENTIFICATION\                         ;
                                          FF  000E3          .BYTE   -1                          ;
                                              000E4 ;TPA$KEYSTO
                                                    U.220:   .BLKB   0
45 49 46 49 54 4E 45 44 49 5F 52 45 4E 57 4F  000E4 ;TPA$KEYST
                                                    U.222:   .ASCII  \OWNER_IDENTIFIER\          ;
                                          52  000F3                                              ;
                                          FF  000F4          .BYTE   -1                          ;
                                          FF  000F5 ;TPA$KEYFILL
                                                    U.226:   .BYTE   -1                          ;
                                              000F6 ;TPA$KEYSTO
                                                    U.237:   .BLKB   0
                           45 55 51 49 4E 55  000F6 ;TPA$KEYST
                                                    U.239:   .ASCII  \UNIQUE\                    ;
                                          FF  000FC          .BYTE   -1                          ;
                                              000FD ;TPA$KEYSTO
                                                    U.244:   .BLKB   0
                                 45 4D 41 53  000FD ;TPA$KEYST
                                                    U.246:   .ASCII  \SAME\                      ;
                                          FF  00101          .BYTE   -1                          ;
                                          FF  00102 ;TPA$KEYFILL
                                                    U.258:   .BYTE   -1                          ;
                                              00103 ;TPA$KEYSTO
                                                    U.284:   .BLKB   0
                           4D 45 54 53 59 53  00103 ;TPA$KEYST
                                                    U.286:   .ASCII  \SYSTEM\                    ;
                                          FF  00109          .BYTE   -1                          ;
                                              0010A ;TPA$KEYSTO
                                                    U.292:   .BLKB   0
                              52 45 4E 57 4F  0010A ;TPA$KEYST
                                                    U.294:   .ASCII  \OWNER\                     ;
                                          FF  0010F          .BYTE   -1                          ;
                                              00110 ;TPA$KEYSTO
                                                    U.300:   .BLKB   0
                              50 55 4F 52 47  00110 ;TPA$KEYST
                                                    U.302:   .ASCII  \GROUP\                     ;
                                          FF  00115          .BYTE   -1                          ;
                                              00116 ;TPA$KEYSTO
                                                    U.308:   .BLKB   0
                              44 4C 52 4F 57  00116 ;TPA$KEYST
                                                    U.310:   .ASCII  \WORLD\                     ;
                                          FF  0011B          .BYTE   -1                          ;
                                          FF  0011C ;TPA$KEYFILL
                                                    U.316:   .BYTE   -1                          ;

                                                             .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                                              00000 CACHE_STB::
                                                             .BLKB   0
                                              00000 NEXT_CACHE:
                                                             .BLKB   0
                                        7100  00000 ;TPA$TYPE
                                                    U.5:     .WORD   28928                       ;
                                   00000000*  00002 ;TPA$ADDR
                                                    U.6:     .LONG   <<<MOUNT_OPTIONS+4>-U.6>-4> ;
                                    00002000  00006 ;TPA$MASK
                                                    U.7:     .LONG   8192                        ;
                                       0000*  0000A ;TPA$TARGET
```

```
                          U.9:      .WORD   <<U.8-U.9>-2>                                   ;
     7101  0000C   ;TPA$TYPE
                          U.13:     .WORD   28929                                           ;
00000000*  0000E   ;TPA$ADDR
                          U.14:     .LONG   <<<MOUNT_OPTIONS+4>-U.14>-4>                    ;
00002000  00012   ;TPA$MASK
                          U.15:     .LONG   8192                                            ;
     0000*  00016   ;TPA$TARGET
                          U.17:     .WORD   <<U.16-U.17>-2>                                 ;
     1102  00018   ;TPA$TYPE
                          U.21:     .WORD   4354                                            ;
     0000*  0001A   ;TPA$TARGET
                          U.23:     .WORD   <<U.22-U.23>-2>                                 ;
     6103  0001C   ;TPA$TYPE
                          U.27:     .WORD   24835                                           ;
00000000*  0001E   ;TPA$ADDR
                          U.28:     .LONG   <<<MOUNT_OPTIONS+4>-U.28>-4>                    ;
00008000  00022   ;TPA$MASK
                          U.29:     .LONG   32768                                           ;
     6104  00026   ;TPA$TYPE
                          U.33:     .WORD   24836                                           ;
00000000*  00028   ;TPA$ADDR
                          U.34:     .LONG   <<<MOUNT_OPTIONS+4>-U.34>-4>                    ;
00010000  0002C   ;TPA$MASK
                          U.35:     .LONG   65536                                           ;
     6105  00030   ;TPA$TYPE
                          U.39:     .WORD   24837                                           ;
00000000*  00032   ;TPA$ADDR
                          U.40:     .LONG   <<<MOUNT_OPTIONS+4>-U.40>-4>                    ;
00020000  00036   ;TPA$MASK
                          U.41:     .LONG   131072                                          ;
     0106  0003A   ;TPA$TYPE
                          U.45:     .WORD   262                                             ;
     7107  0003C   ;TPA$TYPE
                          U.49:     .WORD   28935                                           ;
00000000*  0003E   ;TPA$ADDR
                          U.50:     .LONG   <<<MOUNT_OPTIONS+4>-U.50>-4>                    ;
00002000  00042   ;TPA$MASK
                          U.51:     .LONG   8192                                            ;
     0000*  00046   ;TPA$TARGET
                          U.53:     .WORD   <<U.52-U.53>-2>                                 ;
     6508  00048   ;TPA$TYPE
                          U.57:     .WORD   25864                                           ;
00000000*  0004A   ;TPA$ADDR
                          U.58:     .LONG   <<<MOUNT_OPTIONS+4>-U.58>-4>                    ;
00004000  0004E   ;TPA$MASK
                          U.59:     .LONG   16384                                           ;
           00052   END_CACHE:
                                    .BLKB   0
     102C  00052   ;TPA$TYPE
                          U.61:     .WORD   4140                                            ;
     0000*  00054   ;TPA$TARGET
                          U.62:     .WORD   <<NEXT_CACHE-U.62>-2>                           ;
     15F7  00056   ;TPA$TYPE
                          U.63:     .WORD   5623                                            ;
     FFFF  00058   ;TPA$TARGET
                          U.64:     .WORD   -1                                              ;
```

```
                0005A ;CACHE_EXT
                      U.8:      .BLKB   0
          003A  0005A ;TPA$TYPE
                      U.65:     .WORD   58                                              ;
          043D  0005C ;TPA$TYPE
                      U.66:     .WORD   1085                                            ;
          55F3  0005E ;TPA$TYPE
                      U.67:     .WORD   22003                                           ;
     00000000*  00060 ;TPA$ADDR
                      U.68:     .LONG   <<EXT_CACHE-U.68>-4>                            ;
         0000*  00064 ;TPA$TARGET
                      U.69:     .WORD   <<END_CACHE-U.69>-2>                            ;
                00066 ;CACHE_FID
                      U.16:     .BLKB   0
          003A  00066 ;TPA$TYPE
                      U.70:     .WORD   58                                              ;
          043D  00068 ;TPA$TYPE
                      U.71:     .WORD   1085                                            ;
          55F3  0006A ;TPA$TYPE
                      U.72:     .WORD   22003                                           ;
     00000000*  0006C ;TPA$ADDR
                      U.73:     .LONG   <<FID_CACHE-U.73>-4>                            ;
         0000*  00070 ;TPA$TARGET
                      U.74:     .WORD   <<END_CACHE-U.74>-2>                            ;
                00072 ;CACHE_QUO
                      U.52:     .BLKB   0
          003A  00072 ;TPA$TYPE
                      U.75:     .WORD   58                                              ;
          043D  00074 ;TPA$TYPE
                      U.76:     .WORD   1085                                            ;
          55F3  00076 ;TPA$TYPE
                      U.77:     .WORD   22003                                           ;
     00000000*  00078 ;TPA$ADDR
                      U.78:     .LONG   <<QUO_CACHE-U.78>-4>                            ;
         0000*  0007C ;TPA$TARGET
                      U.79:     .WORD   <<END_CACHE-U.79>-2>                            ;
                0007E ;LIMIT_EXT
                      U.22:     .BLKB   0
          003A  0007E ;TPA$TYPE
                      U.80:     .WORD   58                                              ;
          043D  00080 ;TPA$TYPE
                      U.81:     .WORD   1085                                            ;
          55F3  00082 ;TPA$TYPE
                      U.82:     .WORD   22003                                           ;
     00000000*  00084 ;TPA$ADDR
                      U.83:     .LONG   <<EXT_LIMIT-U.83>-4>                            ;
         0000*  00088 ;TPA$TARGET
                      U.84:     .WORD   <<END_CACHE-U.84>-2>                            ;
                0008A           .BLKB   2
                0008C DATACHECK_STB::
                                .BLKB   0
          71F7  0008C ;TPA$TYPE
                      U.86:     .WORD   29175                                           ;
     00000000*  0008E ;TPA$ADDR
                      U.87:     .LONG   <<<MOUNT_OPTIONS+4>-U.87>-4>                    ;
      00000010  00092 ;TPA$MASK
                      U.88:     .LONG   16                                              ;
         FFFF  00096 ;TPA$TARGET
                     U.89:    .WORD   -1                                   ;
         05F6  00098 ;TPA$TYPE
                     U.90:    .WORD   1526                                 ;
               0009A CHECKOPT:
                              .BLKB   0
         6100  0009A ;TPA$TYPE
                     U.94:    .WORD   24832                                ;
    00000000*  0009C ;TPA$ADDR
                     U.95:    .LONG   <<<MOUNT_OPTIONS+4>-U.95>-4>         ;
     00000008  000A0 ;TPA$MASK
                     U.96:    .LONG   8                                    ;
         6501  000A4 ;TPA$TYPE
                     U.100:   .WORD   25857                                ;
    00000000*  000A6 ;TPA$ADDR
                     U.101:   .LONG   <<<MOUNT_OPTIONS+4>-U.101>-4>        ;
     00000010  000AA ;TPA$MASK
                     U.102:   .LONG   16                                   ;
         102C  000AE ;TPA$TYPE
                     U.104:   .WORD   4140                                 ;
        0000*  000B0 ;TPA$TARGET
                     U.105:   .WORD   <<CHECKOPT-U.105>-2>                 ;
         15F7  000B2 ;TPA$TYPE
                     U.106:   .WORD   5623                                 ;
         FFFF  000B4 ;TPA$TARGET
                     U.107:   .WORD   -1                                   ;
               000B6          .BLKB   2
               000B8 INITIALIZE_STB::
                              .BLKB   0
               000B8 NEXTINI:.BLKB    0
         6100  000B8 ;TPA$TYPE
                     U.112:   .WORD   24832                                ;
    00000000*  000BA ;TPA$ADDR
                     U.113:   .LONG   <<<MOUNT_OPTIONS+4>-U.113>-4>        ;
     04000000  000BE ;TPA$MASK
                     U.114:   .LONG   67108864                             ;
         6501  000C2 ;TPA$TYPE
                     U.118:   .WORD   25857                                ;
    00000000*  000C4 ;TPA$ADDR
                     U.119:   .LONG   <<<MOUNT_OPTIONS+4>-U.119>-4>        ;
     08000000  000C8 ;TPA$MASK
                     U.120:   .LONG   134217728                            ;
         102C  000CC ;TPA$TYPE
                     U.122:   .WORD   4140                                 ;
        0000*  000CE ;TPA$TARGET
                     U.123:   .WORD   <<NEXTINI-U.123>-2>                  ;
         15F7  000D0 ;TPA$TYPE
                     U.124:   .WORD   5623                                 ;
         FFFF  000D2 ;TPA$TARGET
                     U.125:   .WORD   -1                                   ;
               000D4 JOURNAL_STB::
                              .BLKB   0
               000D4 NEXT_JOURNAL:
                              .BLKB   0
         6100  000D4 ;TPA$TYPE
                     U.130:   .WORD   24832                                ;
    00000000*  000D6 ;TPA$ADDR
```

```
                          U.131:   .LONG    <<<MOUNT_OPTIONS+4>-U.131>-4>                 ;
 01000000  000DA ;TPA$MASK
                          U.132:   .LONG    16777216                                      ;
     8101  000DE ;TPA$TYPE
                          U.136:   .WORD    -32511                                        ;
00000000*  000E0 ;TPA$ACTION
                          U.137:   .LONG    <<CLEAR_NEWJRNL-U.137>-4>                     ;
     1102  000E4 ;TPA$TYPE
                          U.141:   .WORD    4354                                          ;
    0000*  000E6 ;TPA$TARGET
                          U.143:   .WORD    <<U.142-U.143>-2>                             ;
     1103  000E8 ;TPA$TYPE
                          U.147:   .WORD    4355                                          ;
    0000*  000EA ;TPA$TARGET
                          U.149:   .WORD    <<U.148-U.149>-2>                             ;
     1104  000EC ;TPA$TYPE
                          U.153:   .WORD    4356                                          ;
    0000*  000EE ;TPA$TARGET
                          U.155:   .WORD    <<U.154-U.155>-2>                             ;
     1105  000F0 ;TPA$TYPE
                          U.159:   .WORD    4357                                          ;
    0000*  000F2 ;TPA$TARGET
                          U.161:   .WORD    <<U.160-U.161>-2>                             ;
     15F7  000F4 ;TPA$TYPE
                          U.162:   .WORD    5623                                          ;
     FFFF  000F6 ;TPA$TARGET
                          U.163:   .WORD    -1                                            ;
           000F8 END_JOURNAL:
                                   .BLKB    0
     102C  000F8 ;TPA$TYPE
                          U.165:   .WORD    4140                                          ;
    0000*  000FA ;TPA$TARGET
                          U.166:   .WORD    <<NEXT_JOURNAL-U.166>-2>                      ;
     15F7  000FC ;TPA$TYPE
                          U.167:   .WORD    5623                                          ;
     FFFF  000FE ;TPA$TARGET
                          U.168:   .WORD    -1                                            ;
           00100 ;JOURNAL_SIZE
                          U.142:   .BLKB    0
     003A  00100 ;TPA$TYPE
                          U.169:   .WORD    58                                            ;
     043D  00102 ;TPA$TYPE
                          U.170:   .WORD    1085                                          ;
     55F3  00104 ;TPA$TYPE
                          U.171:   .WORD    22003                                         ;
00000000*  00106 ;TPA$ADDR
                          U.172:   .LONG    <<JRNL_SIZE-U.172>-4>                         ;
    0000*  0010A ;TPA$TARGET
                          U.173:   .WORD    <<END_JOURNAL-U.173>-2>                       ;
           0010C ;JOURNAL_RECORD_SIZE
                          U.148:   .BLKB    0
     003A  0010C ;TPA$TYPE
                          U.174:   .WORD    58                                            ;
     043D  0010E ;TPA$TYPE
                          U.175:   .WORD    1085                                          ;
     55F3  00110 ;TPA$TYPE
                          U.176:   .WORD    22003                                         ;
```

```
00000000* 00112 ;TPA$ADDR
                U.177:  .LONG   <<JRNL_RECORD_SIZE-U.177>-4>              ;
    0000* 00116 ;TPA$TARGET
                U.178:  .WORD   <<END_JOURNAL-U.178>-2>                   ;
          00118 ;JOURNAL_EXTEND
                U.154:  .BLKB   0
    003A  00118 ;TPA$TYPE
                U.179:  .WORD   58                                        ;
    043D  0011A ;TPA$TYPE
                U.180:  .WORD   1085                                      ;
    55F3  0011C ;TPA$TYPE
                U.181:  .WORD   22003                                     ;
00000000* 0011E ;TPA$ADDR
                U.182:  .LONG   <<JRNL_EXTEND-U.182>-4>                   ;
    0000* 00122 ;TPA$TARGET
                U.183:  .WORD   <<END_JOURNAL-U.183>-2>                   ;
          00124 ;JOURNAL_QUOTA
                U.160:  .BLKB   0
    003A  00124 ;TPA$TYPE
                U.184:  .WORD   58                                        ;
    043D  00126 ;TPA$TYPE
                U.185:  .WORD   1085                                      ;
    55F3  00128 ;TPA$TYPE
                U.186:  .WORD   22003                                     ;
00000000* 0012A ;TPA$ADDR
                U.187:  .LONG   <<JRNL_QUOTA-U.187>-4>                    ;
    0000* 0012E ;TPA$TARGET
                U.188:  .WORD   <<END_JOURNAL-U.188>-2>                   ;
          00130 OVERRIDE_STB::
                        .BLKB   0
          00130 NEXTOVR:.BLKB   0
    6100  00130 ;TPA$TYPE
                U.193:  .WORD   24832                                     ;
00000000* 00132 ;TPA$ADDR
                U.194:  .LONG   <<<MOUNT_OPTIONS+4>-U.194>-4>             ;
00000040  00136 ;TPA$MASK
                U.195:  .LONG   64                                        ;
    6101  0013A ;TPA$TYPE
                U.199:  .WORD   24833                                     ;
00000000* 0013C ;TPA$ADDR
                U.200:  .LONG   <<MOUNT_OPTIONS-U.200>-4>                 ;
00100000  00140 ;TPA$MASK
                U.201:  .LONG   1048576                                   ;
    6102  00144 ;TPA$TYPE
                U.205:  .WORD   24834                                     ;
00000000* 00146 ;TPA$ADDR
                U.206:  .LONG   <<MOUNT_OPTIONS-U.206>-4>                 ;
00200000  0014A ;TPA$MASK
                U.207:  .LONG   2097152                                   ;
    6103  0014E ;TPA$TYPE
                U.211:  .WORD   24835                                     ;
00000000* 00150 ;TPA$ADDR
                U.212:  .LONG   <<<MOUNT_OPTIONS+4>-U.212>-4>             ;
00200000  00154 ;TPA$MASK
                U.213:  .LONG   2097152                                   ;
    6104  00158 ;TPA$TYPE
                U.217:  .WORD   24836                                     ;
```

```
00000000* 0015A ;TPA$ADDR
                U.218:    .LONG   <<MOUNT_OPTIONS-U.218>-4>                 ;
00400000  0015E ;TPA$MASK
                U.219:    .LONG   4194304                                   ;
    6505  00162 ;TPA$TYPE
                U.223:    .WORD   25861                                     ;
00000000* 00164 ;TPA$ADDR
                U.224:    .LONG   <<<MOUNT_OPTIONS+4>-U.224>-4>             ;
10000000  00168 ;TPA$MASK
                U.225:    .LONG   268435456                                 ;
    102C  0016C ;TPA$TYPE
                U.227:    .WORD   4140                                      ;
   0000*  0016E ;TPA$TARGET
                U.228:    .WORD   <<NEXTOVR-U.228>-2>                       ;
    15F7  00170 ;TPA$TYPE
                U.229:    .WORD   5623                                      ;
    FFFF  00172 ;TPA$TARGET
                U.230:    .WORD   -1                                        ;
          00174 UIC_STB::
                          .BLKB   0
    45EC  00174 ;TPA$TYPE
                U.232:    .WORD   17900                                     ;
00000000* 00176 ;TPA$ADDR
                U.233:    .LONG   <<UIC-U.233>-4>                           ;
    15F7  0017A ;TPA$TYPE
                U.234:    .WORD   5623                                      ;
    FFFF  0017C ;TPA$TARGET
                U.235:    .WORD   -1                                        ;
          0017E           .BLKB   2
          00180 PROCESSOR_STB::
                          .BLKB   0
    E100  00180 ;TPA$TYPE
                U.240:    .WORD   -7936                                     ;
00000000* 00182 ;TPA$ACTION
                U.241:    .LONG   <<GET_ACP_NAME-U.241>-4>                  ;
00000000* 00186 ;TPA$ADDR
                U.242:    .LONG   <<MOUNT_OPTIONS-U.242>-4>                 ;
04000000  0018A ;TPA$MASK
                U.243:    .LONG   67108864                                  ;
    7101  0018E ;TPA$TYPE
                U.247:    .WORD   28929                                     ;
00000000* 00190 ;TPA$ADDR
                U.248:    .LONG   <<MOUNT_OPTIONS-U.248>-4>                 ;
08000000  00194 ;TPA$MASK
                U.249:    .LONG   134217728                                 ;
   0000*  00198 ;TPA$TARGET
                U.251:    .WORD   <<U.250-U.251>-2>                         ;
    EDF8  0019A ;TPA$TYPE
                U.252:    .WORD   -4616                                     ;
   0000*  0019C ;TPA$SUBEXP
                U.254:    .WORD   <<U.253-U.254>-2>                         ;
00000000* 0019E ;TPA$ACTION
                U.255:    .LONG   <<GET_ACP_NAME-U.255>-4>                  ;
00000000* 001A2 ;TPA$ADDR
                U.256:    .LONG   <<MOUNT_OPTIONS-U.256>-4>                 ;
10000000  001A6 ;TPA$MASK
                U.257:    .LONG   268435456                                 ;
```

```
             001AA ENDPROC::.BLKB    0
     15F7    001AA ;TPA$TYPE
                   U.259:   .WORD    5623                                         ;
     FFFF    001AC ;TPA$TARGET
                   U.260:   .WORD    -1                                           ;
             001AE ;SAMEPROC
                   U.250:   .BLKB    0
     003A    001AE ;TPA$TYPE
                   U.261:   .WORD    58                                           ;
     043D    001B0 ;TPA$TYPE
                   U.262:   .WORD    1085                                         ;
     89FB    001B2 ;TPA$TYPE
                   U.263:   .WORD    -30216                                       ;
    0000*    001B4 ;TPA$SUBEXP
                   U.265:   .WORD    <<U.264-U.265>-2>                            ;
00000000*    001B6 ;TPA$ACTION
                   U.266:   .LONG    <<GET_ACP_NAME-U.266>-4>                     ;
     85F1    001BA ;TPA$TYPE
                   U.267:   .WORD    -31247                                       ;
00000000*    001BC ;TPA$ACTION
                   U.268:   .LONG    <<GET_SAME_ACP-U.268>-4>                     ;
     15F6    001C0 ;TPA$TYPE
                   U.269:   .WORD    5622                                         ;
     FFFF    001C2 ;TPA$TARGET
                   U.270:   .WORD    -1                                           ;
             001C4 ;FILENAME
                   U.253:   .BLKB    0
     11F1    001C4 ;TPA$TYPE
                   U.271:   .WORD    4593                                         ;
    0000*    001C6 ;TPA$TARGET
                   U.272:   .WORD    <<U.253-U.272>-2>                            ;
     102E    001C8 ;TPA$TYPE
                   U.273:   .WORD    4142                                         ;
    0000*    001CA ;TPA$TARGET
                   U.274:   .WORD    <<U.253-U.274>-2>                            ;
     103B    001CC ;TPA$TYPE
                   U.275:   .WORD    4155                                         ;
    0000*    001CE ;TPA$TARGET
                   U.276:   .WORD    <<U.253-U.276>-2>                            ;
     15F6    001D0 ;TPA$TYPE
                   U.277:   .WORD    5622                                         ;
     FFFF    001D2 ;TPA$TARGET
                   U.278:   .WORD    -1                                           ;
             001D4 ;DEVICENAME
                   U.264:   .BLKB    0
     05F1    001D4 ;TPA$TYPE
                   U.279:   .WORD    1521                                         ;
     043A    001D6 ;TPA$TYPE
                   U.280:   .WORD    1082                                         ;
     15F7    001D8 ;TPA$TYPE
                   U.281:   .WORD    5623                                         ;
     FFFF    001DA ;TPA$TARGET
                   U.282:   .WORD    -1                                           ;
             001DC PROTECTION_STB::
                            .BLKB    0
             001DC NEXTPRO::.BLKB    0
     7100    001DC ;TPA$TYPE
```

```
                          U.287:   .WORD     28928                                ;
     00000000*   001DE   ;TPA$ADDR
                          U.288:   .LONG     <<PROTECTION-U.288>-4>               ;
     000F0000    001E2   ;TPA$MASK
                          U.289:   .LONG     983040                               ;
         0000*   001E6   ;TPA$TARGET
                          U.291:   .WORD     <<U.290-U.291>-2>                    ;
         7101    001E8   ;TPA$TYPE
                          U.295:   .WORD     28929                                ;
     00000000*   001EA   ;TPA$ADDR
                          U.296:   .LONG     <<PROTECTION-U.296>-4>               ;
     00F00000    001EE   ;TPA$MASK
                          U.297:   .LONG     15728640                             ;
         0000*   001F2   ;TPA$TARGET
                          U.299:   .WORD     <<U.298-U.299>-2>                    ;
         7102    001F4   ;TPA$TYPE
                          U.303:   .WORD     28930                                ;
     00000000*   001F6   ;TPA$ADDR
                          U.304:   .LONG     <<PROTECTION-U.304>-4>               ;
     0F000000    001FA   ;TPA$MASK
                          U.305:   .LONG     251658240                            ;
         0000*   001FE   ;TPA$TARGET
                          U.307:   .WORD     <<U.306-U.307>-2>                    ;
         7503    00200   ;TPA$TYPE
                          U.311:   .WORD     29955                                ;
     00000000*   00202   ;TPA$ADDR
                          U.312:   .LONG     <<PROTECTION-U.312>-4>               ;
     F0000000    00206   ;TPA$MASK
                          U.313:   .LONG     -268435456                           ;
         0000*   0020A   ;TPA$TARGET
                          U.315:   .WORD     <<U.314-U.315>-2>                    ;
                 0020C   ;SYPR
                          U.290:   .BLKB     0
         003A    0020C   ;TPA$TYPE
                          U.317:   .WORD     58                                   ;
         003D    0020E   ;TPA$TYPE
                          U.318:   .WORD     61                                   ;
         15F6    00210   ;TPA$TYPE
                          U.319:   .WORD     5622                                 ;
         0000*   00212   ;TPA$TARGET
                          U.321:   .WORD     <<U.320-U.321>-2>                    ;
                 00214   SYPRO:    .BLKB     0
         7052    00214   ;TPA$TYPE
                          U.322:   .WORD     28754                                ;
     00000000*   00216   ;TPA$ADDR
                          U.323:   .LONG     <<PROTECTION-U.323>-4>               ;
     00000001    0021A   ;TPA$MASK
                          U.324:   .LONG     1                                    ;
         0000*   0021E   ;TPA$TARGET
                          U.325:   .WORD     <<SYPRO-U.325>-2>                    ;
         7057    00220   ;TPA$TYPE
                          U.326:   .WORD     28759                                ;
     00000000*   00222   ;TPA$ADDR
                          U.327:   .LONG     <<PROTECTION-U.327>-4>               ;
     00000002    00226   ;TPA$MASK
                          U.328:   .LONG     2                                    ;
         0000*   0022A   ;TPA$TARGET
```

```
                         U.329:   .WORD    <<SYPRO-U.329>-2>                    ;
     7045   0022C  ;TPA$TYPE
                         U.330:   .WORD    28741                                ;
00000000*   0022E  ;TPA$ADDR
                         U.331:   .LONG    <<PROTECTION-U.331>-4>               ;
00000004    00232  ;TPA$MASK
                         U.332:   .LONG    4                                    ;
    0000*   00236  ;TPA$TARGET
                         U.333:   .WORD    <<SYPRO-U.333>-2>                    ;
     7050   00238  ;TPA$TYPE
                         U.334:   .WORD    28752                                ;
00000000*   0023A  ;TPA$ADDR
                         U.335:   .LONG    <<PROTECTION-U.335>-4>               ;
00000004    0023E  ;TPA$MASK
                         U.336:   .LONG    4                                    ;
    0000*   00242  ;TPA$TARGET
                         U.337:   .WORD    <<SYPRO-U.337>-2>                    ;
     7044   00244  ;TPA$TYPE
                         U.338:   .WORD    28740                                ;
00000000*   00246  ;TPA$ADDR
                         U.339:   .LONG    <<PROTECTION-U.339>-4>               ;
00000008    0024A  ;TPA$MASK
                         U.340:   .LONG    8                                    ;
    0000*   0024E  ;TPA$TARGET
                         U.341:   .WORD    <<SYPRO-U.341>-2>                    ;
     704C   00250  ;TPA$TYPE
                         U.342:   .WORD    28748                                ;
00000000*   00252  ;TPA$ADDR
                         U.343:   .LONG    <<PROTECTION-U.343>-4>               ;
00000008    00256  ;TPA$MASK
                         U.344:   .LONG    8                                    ;
    0000*   0025A  ;TPA$TARGET
                         U.345:   .WORD    <<SYPRO-U.345>-2>                    ;
     15F6   0025C  ;TPA$TYPE
                         U.346:   .WORD    5622                                 ;
    0000*   0025E  ;TPA$TARGET
                         U.347:   .WORD    <<U.320-U.347>-2>                    ;
            00260  ;OWPR
                         U.298:   .BLKB    0
     003A   00260  ;TPA$TYPE
                         U.348:   .WORD    58                                   ;
     003D   00262  ;TPA$TYPE
                         U.349:   .WORD    61                                   ;
     15F6   00264  ;TPA$TYPE
                         U.350:   .WORD    5622                                 ;
    0000*   00266  ;TPA$TARGET
                         U.351:   .WORD    <<U.320-U.351>-2>                    ;
            00268  OWPRO:  .BLKB   0
     7052   00268  ;TPA$TYPE
                         U.352:   .WORD    28754                                ;
00000000*   0026A  ;TPA$ADDR
                         U.353:   .LONG    <<PROTECTION-U.353>-4>               ;
00000010    0026E  ;TPA$MASK
                         U.354:   .LONG    16                                   ;
    0000*   00272  ;TPA$TARGET
                         U.355:   .WORD    <<OWPRO-U.355>-2>                    ;
     7057   00274  ;TPA$TYPE
```

```
                 U.356:   .WORD    28759                                          ;
00000000* 00276 ;TPA$ADDR
                 U.357:   .LONG    <<PROTECTION-U.357>-4>                         ;
00000020  0027A ;TPA$MASK
                 U.358:   .LONG    32                                             ;
    0000* 0027E ;TPA$TARGET
                 U.359:   .WORD    <<OWPRO-U.359>-2>                              ;
    7045  00280 ;TPA$TYPE
                 U.360:   .WORD    28741                                          ;
00000000* 00282 ;TPA$ADDR
                 U.361:   .LONG    <<PROTECTION-U.361>-4>                         ;
00000040  00286 ;TPA$MASK
                 U.362:   .LONG    64                                             ;
    0000* 0028A ;TPA$TARGET
                 U.363:   .WORD    <<OWPRO-U.363>-2>                              ;
    7050  0028C ;TPA$TYPE
                 U.364:   .WORD    28752                                          ;
00000000* 0028E ;TPA$ADDR
                 U.365:   .LONG    <<PROTECTION-U.365>-4>                         ;
00000040  00292 ;TPA$MASK
                 U.366:   .LONG    64                                             ;
    0000* 00296 ;TPA$TARGET
                 U.367:   .WORD    <<OWPRO-U.367>-2>                              ;
    7044  00298 ;TPA$TYPE
                 U.368:   .WORD    28740                                          ;
00000000* 0029A ;TPA$ADDR
                 U.369:   .LONG    <<PROTECTION-U.369>-4>                         ;
00000080  0029E ;TPA$MASK
                 U.370:   .LONG    128                                            ;
    0000* 002A2 ;TPA$TARGET
                 U.371:   .WORD    <<OWPRO-U.371>-2>                              ;
    704C  002A4 ;TPA$TYPE
                 U.372:   .WORD    28748                                          ;
00000000* 002A6 ;TPA$ADDR
                 U.373:   .LONG    <<PROTECTION-U.373>-4>                         ;
00000080  002AA ;TPA$MASK
                 U.374:   .LONG    128                                            ;
    0000* 002AE ;TPA$TARGET
                 U.375:   .WORD    <<OWPRO-U.375>-2>                              ;
    15F6  002B0 ;TPA$TYPE
                 U.376:   .WORD    5622                                           ;
    0000* 002B2 ;TPA$TARGET
                 U.377:   .WORD    <<U.320-U.377>-2>                              ;
          002B4 ;GRPR
                 U.306:   .BLKB    0
    003A  002B4 ;TPA$TYPE
                 U.378:   .WORD    58                                             ;
    003D  002B6 ;TPA$TYPE
                 U.379:   .WORD    61                                             ;
    15F6  002B8 ;TPA$TYPE
                 U.380:   .WORD    5622                                           ;
    0000* 002BA ;TPA$TARGET
                 U.381:   .WORD    <<U.320-U.381>-2>                              ;
          002BC GRPRO:    .BLKB    0
    7052  002BC ;TPA$TYPE
                 U.382:   .WORD    28754                                          ;
00000000* 002BE ;TPA$ADDR
```

```
                        U.383:   .LONG    <<PROTECTION-U.383>-4>              ;
  00000100  002C2  ;TPA$MASK
                        U.384:   .LONG    256                                 ;
     0000*  002C6  ;TPA$TARGET
                        U.385:   .WORD    <<GRPRO-U.385>-2>                   ;
      7057  002C8  ;TPA$TYPE
                        U.386:   .WORD    28759                               ;
 00000000*  002CA  ;TPA$ADDR
                        U.387:   .LONG    <<PROTECTION-U.387>-4>              ;
  00000200  002CE  ;TPA$MASK
                        U.388:   .LONG    512                                 ;
     0000*  002D2  ;TPA$TARGET
                        U.389:   .WORD    <<GRPRO-U.389>-2>                   ;
      7045  002D4  ;TPA$TYPE
                        U.390:   .WORD    28741                               ;
 00000000*  002D6  ;TPA$ADDR
                        U.391:   .LONG    <<PROTECTION-U.391>-4>              ;
  00000400  002DA  ;TPA$MASK
                        U.392:   .LONG    1024                                ;
     0000*  002DE  ;TPA$TARGET
                        U.393:   .WORD    <<GRPRO-U.393>-2>                   ;
      7050  002E0  ;TPA$TYPE
                        U.394:   .WORD    28752                               ;
 00000000*  002E2  ;TPA$ADDR
                        U.395:   .LONG    <<PROTECTION-U.395>-4>              ;
  00000400  002E6  ;TPA$MASK
                        U.396:   .LONG    1024                                ;
     0000*  002EA  ;TPA$TARGET
                        U.397:   .WORD    <<GRPRO-U.397>-2>                   ;
      7044  002EC  ;TPA$TYPE
                        U.398:   .WORD    28740                               ;
 00000000*  002EE  ;TPA$ADDR
                        U.399:   .LONG    <<PROTECTION-U.399>-4>              ;
  00000800  002F2  ;TPA$MASK
                        U.400:   .LONG    2048                                ;
     0000*  002F6  ;TPA$TARGET
                        U.401:   .WORD    <<GRPRO-U.401>-2>                   ;
      704C  002F8  ;TPA$TYPE
                        U.402:   .WORD    28748                               ;
 00000000*  002FA  ;TPA$ADDR
                        U.403:   .LONG    <<PROTECTION-U.403>-4>              ;
  00000800  002FE  ;TPA$MASK
                        U.404:   .LONG    2048                                ;
     0000*  00302  ;TPA$TARGET
                        U.405:   .WORD    <<GRPRO-U.405>-2>                   ;
      15F6  00304  ;TPA$TYPE
                        U.406:   .WORD    5622                                ;
     0000*  00306  ;TPA$TARGET
                        U.407:   .WORD    <<U.320-U.407>-2>                   ;
            00308  ;WOPR
                        U.314:   .BLKB    0
      003A  00308  ;TPA$TYPE
                        U.408:   .WORD    58                                  ;
      003D  0030A  ;TPA$TYPE
                        U.409:   .WORD    61                                  ;
      15F6  0030C  ;TPA$TYPE
                        U.410:   .WORD    5622                                ;
```

```
       0000* 0030E ;TPA$TARGET
                   U.411:   .WORD   <<U.320-U.411>-2>                          ;
             00310 WOPRO:   .BLKB   0
       7052  00310 ;TPA$TYPE
                   U.412:   .WORD   28754                                      ;
   00000000* 00312 ;TPA$ADDR
                   U.413:   .LONG   <<PROTECTION-U.413>-4>                     ;
   00001000  00316 ;TPA$MASK
                   U.414:   .LONG   4096                                       ;
       0000* 0031A ;TPA$TARGET
                   U.415:   .WORD   <<WOPRO-U.415>-2>                          ;
       7057  0031C ;TPA$TYPE
                   U.416:   .WORD   28759                                      ;
   00000000* 0031E ;TPA$ADDR
                   U.417:   .LONG   <<PROTECTION-U.417>-4>                     ;
   00002000  00322 ;TPA$MASK
                   U.418:   .LONG   8192                                       ;
       0000* 00326 ;TPA$TARGET
                   U.419:   .WORD   <<WOPRO-U.419>-2>                          ;
       7045  00328 ;TPA$TYPE
                   U.420:   .WORD   28741                                      ;
   00000000* 0032A ;TPA$ADDR
                   U.421:   .LONG   <<PROTECTION-U.421>-4>                     ;
   00004000  0032E ;TPA$MASK
                   U.422:   .LONG   16384                                      ;
       0000* 00332 ;TPA$TARGET
                   U.423:   .WORD   <<WOPRO-U.423>-2>                          ;
       7050  00334 ;TPA$TYPE
                   U.424:   .WORD   28752                                      ;
   00000000* 00336 ;TPA$ADDR
                   U.425:   .LONG   <<PROTECTION-U.425>-4>                     ;
   00004000  0033A ;TPA$MASK
                   U.426:   .LONG   16384                                      ;
       0000* 0033E ;TPA$TARGET
                   U.427:   .WORD   <<WOPRO-U.427>-2>                          ;
       7044  00340 ;TPA$TYPE
                   U.428:   .WORD   28740                                      ;
   00000000* 00342 ;TPA$ADDR
                   U.429:   .LONG   <<PROTECTION-U.429>-4>                     ;
   00008000  00346 ;TPA$MASK
                   U.430:   .LONG   32768                                      ;
       0000* 0034A ;TPA$TARGET
                   U.431:   .WORD   <<WOPRO-U.431>-2>                          ;
       704C  0034C ;TPA$TYPE
                   U.432:   .WORD   28748                                      ;
   00000000* 0034E ;TPA$ADDR
                   U.433:   .LONG   <<PROTECTION-U.433>-4>                     ;
   00008000  00352 ;TPA$MASK
                   U.434:   .LONG   32768                                      ;
       0000* 00356 ;TPA$TARGET
                   U.435:   .WORD   <<WOPRO-U.435>-2>                          ;
       15F6  00358 ;TPA$TYPE
                   U.436:   .WORD   5622                                       ;
       0000* 0035A ;TPA$TARGET
                   U.437:   .WORD   <<U.320-U.437>-2>                          ;
             0035C ;ENDPRO
                   U.320:   .BLKB   0
```

```
102C  0035C ;TPA$TYPE
            U.438:  .WORD   4140                                    ;
0000* 0035E ;TPA$TARGET
            U.439:  .WORD   <<NEXTPRG-U.439>-2>                     ;
15F7  00360 ;TPA$TYPE
            U.440:  .WORD   5623                                    ;
FFFF  00362 ;TPA$TARGET
            U.441:  .WORD   -1                                      ;

                    .PSECT  _LIB$KEY0$,NOWRT,  SHR,  PIC,1

      00000 CACHE_KTB::
                    .BLKB   0
      00000 ;TPA$KEY0
            U.1:    .BLKB   0
0000* 00000 ;TPA$KEY
            U.3:    .WORD   <U.2-U.1>                               ;
0000* 00002 ;TPA$KEY
            U.11:   .WORD   <U.10-U.1>                              ;
0000* 00004 ;TPA$KEY
            U.19:   .WORD   <U.18-U.1>                              ;
0000* 00006 ;TPA$KEY
            U.25:   .WORD   <U.24-U.1>                              ;
0000* 00008 ;TPA$KEY
            U.31:   .WORD   <U.30-U.1>                              ;
0000* 0000A ;TPA$KEY
            U.37:   .WORD   <U.36-U.1>                              ;
0000* 0000C ;TPA$KEY
            U.43:   .WORD   <U.42-U.1>                              ;
0000* 0000E ;TPA$KEY
            U.47:   .WORD   <U.46-U.1>                              ;
0000* 00010 ;TPA$KEY
            U.55:   .WORD   <U.54-U.1>                              ;
      00012         .BLKB   2
      00014 DATACHECK_KTB::
                    .BLKB   0
      00014 ;TPA$KEY0
            U.85:   .BLKB   0
0000* 00014 ;TPA$KEY
            U.92:   .WORD   <U.91-U.85>                             ;
0000* 00016 ;TPA$KEY
            U.98:   .WORD   <U.97-U.85>                             ;
      00018 INITIALIZE_KTB::
                    .BLKB   0
      00018 ;TPA$KEY0
            U.108:  .BLKB   0
0000* 00018 ;TPA$KEY
            U.110:  .WORD   <U.109-U.108>                           ;
0000* 0001A ;TPA$KEY
            U.116:  .WORD   <U.115-U.108>                           ;
      0001C JOURNAL_KTB::
                    .BLKB   0
      0001C ;TPA$KEY0
            U.126:  .BLKB   0
0000* 0001C ;TPA$KEY
            U.128:  .WORD   <U.127-U.126>                           ;
0000* 0001E ;TPA$KEY
```

```
                   U.134:   .WORD    <U.133-U.126>                          ;
0000*  00020 ;TPA$KEY
                   U.139:   .WORD    <U.138-U.126>                          ;
0000*  00022 ;TPA$KEY
                   U.145:   .WORD    <U.144-U.126>                          ;
0000*  00024 ;TPA$KEY
                   U.151:   .WORD    <U.150-U.126>                          ;
0000*  00026 ;TPA$KEY
                   U.157:   .WORD    <U.156-U.126>                          ;
       00028 OVERRIDE_KTB::
                            .BLKB    0
       00028 ;TPA$KEY0
                   U.189:   .BLKB    0
0000*  00028 ;TPA$KEY
                   U.191:   .WORD    <U.190-U.189>                          ;
0000*  0002A ;TPA$KEY
                   U.197:   .WORD    <U.196-U.189>                          ;
0000*  0002C ;TPA$KEY
                   U.203:   .WORD    <U.202-U.189>                          ;
0000*  0002E ;TPA$KEY
                   U.209:   .WORD    <U.208-U.189>                          ;
0000*  00030 ;TPA$KEY
                   U.215:   .WORD    <U.214-U.189>                          ;
0000*  00032 ;TPA$KEY
                   U.221:   .WORD    <U.220-U.189>                          ;
       00034 UIC_KTB::
                            .BLKB    0
       00034 ;TPA$KEY0
                   U.231:   .BLKB    0
       00034 PROCESSOR_KTB::
                            .BLKB    0
       00034 ;TPA$KEY0
                   U.236:   .BLKB    0
0000*  00034 ;TPA$KEY
                   U.238:   .WORD    <U.237-U.236>                          ;
0000*  00036 ;TPA$KEY
                   U.245:   .WORD    <U.244-U.236>                          ;
       00038 PROTECTION_KTB::
                            .BLKB    0
       00038 ;TPA$KEY0
                   U.283:   .BLKB    0
0000*  00038 ;TPA$KEY
                   U.285:   .WORD    <U.284-U.283>                          ;
0000*  0003A ;TPA$KEY
                   U.293:   .WORD    <U.292-U.283>                          ;
0000*  0003C ;TPA$KEY
                   U.301:   .WORD    <U.300-U.283>                          ;
0000*  0003E ;TPA$KEY
                   U.309:   .WORD    <U.308-U.283>                          ;

                            .EXTRN   LIB$STOP

;                         PSECT SUMMARY
;
;      Name                  Bytes                     Attributes
;
```

```
;   $OWN$                              424  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                             600  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                            3426  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   _LIB$KEY0$                          64  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;   _LIB$STATE$                        868  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;   _LIB$KEY1$                         285  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)




;                                 Library Statistics
;
;                                                     -------- Symbols --------       Pages        Processing
;         File                                        Total     Loaded   Percent      Mapped       Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1                     18619        100         0        1000         00:02.0
;   _$255$DUA28:[SYSLIB]CLIMAC.L32;1                     14          0         0           9         00:00.1
;   _$255$DUA28:[SYSLIB]TPAMAC.L32;1                     42         29        69          14         00:00.1





;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MOUNTIMG/OBJ=OBJ$:MOUNTIMG MSRC$:MOUNTIMG/UPDATE=(ENH$:MOUNTIMG)

; Size:           3426 code + 2241 data bytes
; Run Time:          01:52.3
; Elapsed Time:      03:33.8
; Lines/CPU Min:       1337
; Lexemes/CPU-Min: 67996
; Memory Used:  502 pages
; Compilation Complete
```

0245 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY